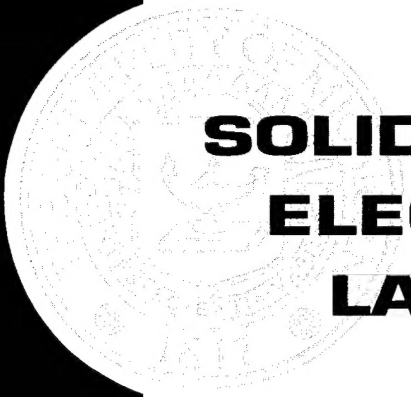
GAAS MESFET STATIC RAM DESIGN FOR EMBEDDED APPLICATIONS

TECHNICAL REPORT NO. SSEL-252

1994

by

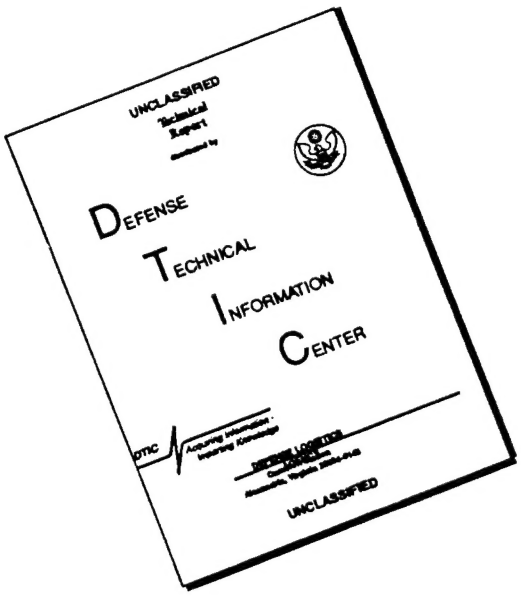
Ajay Chandna



SOLID-STATE ELECTRONICS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 20, 1996	Technical Report

4. TITLE AND SUBTITLE

GaAs MESFET Static RAM Design for Embedded Applications

5. FUNDING NUMBERS

DAAH04-94-G-0327

6. AUTHOR(S)

Ajay Chandna

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

University of Michigan
Dept. of Electrical Engineering & Computer Science
1301 Beal Ave.
Ann Arbor, MI 48109-2122

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 33790.20-EL

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The main focus of this thesis is research in novel GaAs static random access memories for embedded applications. The thesis presents several new circuit structures and design methodologies that are needed to achieve higher performance, lower power, process tolerant static RAMs and digital circuits using E/D MESFETs in GaAs.

High static power dissipation hinders the integration levels of memories that use GaAs E/D MESFETs. This report presents a new logic style, called power rail logic (PRL), that was invented to reduce some of the standby power of inactive circuits. PRL offers circuits with a smaller area and up to 40% lower power-delay products than can be achieved with DCFL. A test chip containing 32-bit DCFL and PRL barrel shifters was designed, fabricated, and tested. The PRL circuit, which was about 12% smaller, was found to operate 13% faster than the DCFL circuit while consuming an average of 24% less power, resulting in a 34% smaller power-delay product.

Finally, this thesis presented the Aurora RAM compiler (ARC) which was developed to generate and characterize highly manufacturable optimized SRAMs using GaAs E/D MESFET technology.

14. SUBJECT TERMS			15. NUMBER IF PAGES
			16. PRICE CODE
17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

This report has also been submitted as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the University of Michigan, 1994.

This work was supported in part by the Advanced Research Projects Agency under Grant DAAH04-94-G-0327.

ABSTRACT

GAAS MESFET STATIC RAM DESIGN FOR EMBEDDED APPLICATIONS

by

Ajay Chandna

Chair: Professor Richard B. Brown

The main focus of this thesis is research in novel GaAs static random access memories for embedded applications. The thesis presents several new circuit structures and design methodologies that are needed to achieve higher performance, lower power, process tolerant static RAMs and digital circuits using E/D MESFETs in GaAs.

Some of the problems that have plagued GaAs E/D MESFET memories are large memory cell sizes, high subthreshold leakage currents, and destructive readout. The current mirror memory cell (CMMC) has been presented as a solution to many of these problems.

High static power dissipation hinders the integration levels of memories that use GaAs E/D MESFETs. This report presents a new logic style, called power rail logic (PRL), that was invented to reduce some of the standby power of inactive circuits. PRL offers circuits with a smaller area and up to 40% lower power-delay products than can be achieved with DCFL. A test chip containing 32-bit DCFL and PRL barrel shifters was designed, fabricated, and tested. The PRL circuit, which was about 12% smaller, was found to operate 13% faster than the DCFL circuit while consuming an average of 24% less power, resulting in a 34% smaller power-delay product.

Finally, this thesis presented the Aurora RAM compiler (ARC) which was developed to generate and characterize highly manufacturable optimized SRAMs using GaAs E/D MESFET technology.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

Rapid progress in silicon VLSI technology has led the way to high levels of integration. Processor clock rates have been increasing at an exponential rate and have recently reached the 200-300 MHz regime in CMOS and Si ECL (see Fig. 1.1). This has placed greater demands on primary level caches to keep pace with processor speeds. As integration levels have increased, architects have begun to exploit the inherent parallelism in instruction streams. Greater than four-way superscalar machines are being designed today. With each execution unit requiring its own instructions and data, this increased parallelism has placed a much greater demand on the bandwidth requirements of primary level instruction and data caches.

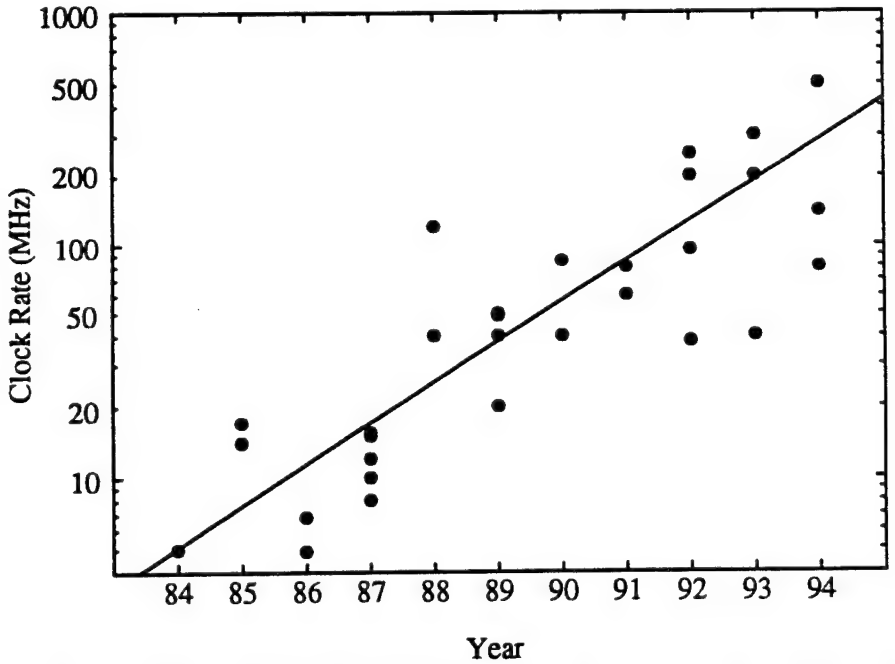


Fig. 1.1: Microprocessor clock rates vs. year presented at ISSCC [Upt93]

Embedded memory has now become a key component of systems on a chip. High performance processors such as the PowerPC 620 are using as much as 64kB of on-chip cache to satisfy their instruction and data needs[Rya94]. It remains to be seen which of the competing technologies, GaAs, CMOS, BiCMOS or Si ECL, will dominate in achieving sub-2ns embedded static random access memories (SRAMs) with greater than 64kB integration densities. The success of any competing high performance digital process technology will lie in its ability to integrate large amounts of high speed memory on chip.

1.1 Trends in SRAM Design

Recent advances in many fields have enabled the fabrication of several high performance SRAMs. The advances that are responsible for these trends include both improved processing technologies and innovative circuit design techniques.

1.1.1 Processing Technology

Advances in processing technologies, such as phase shift lithography, thin film transistors, multi-layer polysilicon and multi-layer aluminum metallization, have led to an ever decreasing cell size for BiCMOS and CMOS research SRAMs. Over the last 5 years, CMOS SRAM cell sizes have undergone a 10-fold reduction in cell area from 21 μm^2 in 1989 to 2.3μm^2 in 1993! Over this time, minimum gate-lengths have seen a reduction of from 0.8μm to 0.25μm. This trend is shown in Fig. 1.2, where cell area is plotted as a function of year of presentation at ISSCC. High density BiCMOS SRAMs use CMOS-only or 4-nMOS/2R cells. The inherent increase in process complexity of BiCMOS seems to have precluded the addition of many features enjoyed by CMOS SRAMs, keeping BiCMOS SRAM cell areas above 18μm^2.

Some of the features of the advanced SRAM processes include up to 5 layers of polysilicon and 3 to 5 layers of metallization. The poly layers are used for many different applications including transistor gates, routing of memory cell power, resistor loads and thin-film transistors.

Clearly, many of the processing techniques that have achieved densities greater than 512kb will not be used for embedded applications. The trend in microprocessor fabrication has

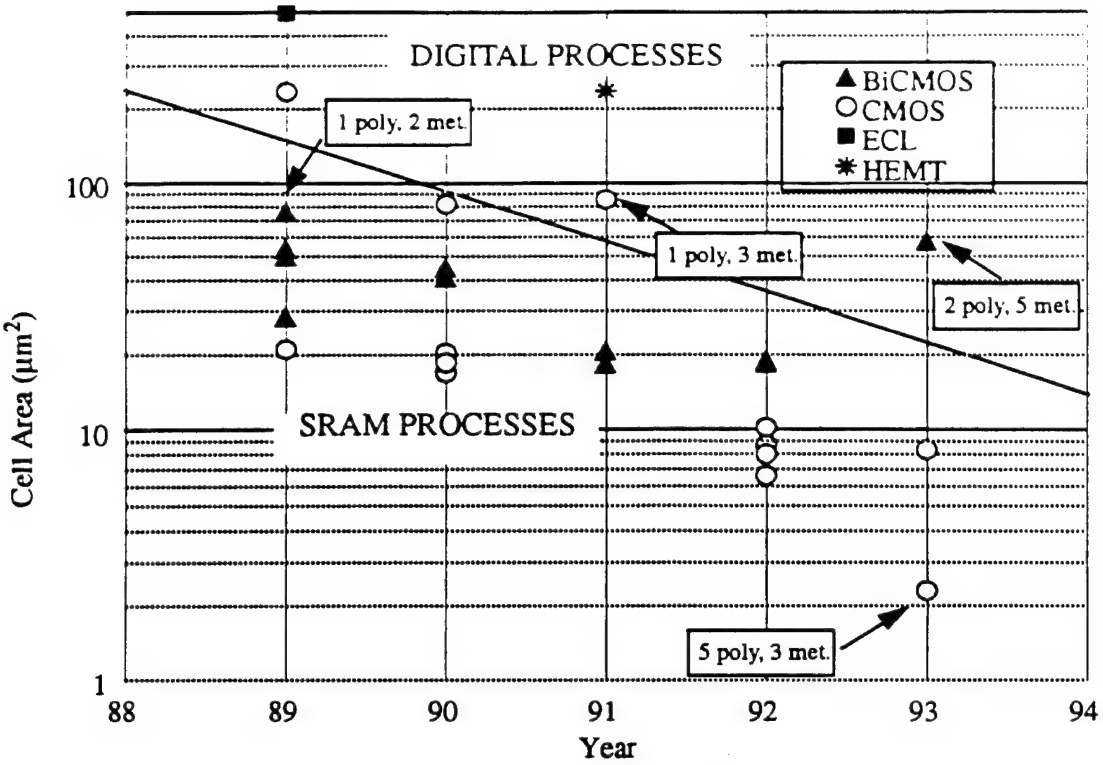


Fig. 1.2: SRAM cell area vs. year presented at ISSCC for CMOS, BiCMOS, ECL, HEMT.

been to stay with conventional digital processes as much as possible for (a) achieving reasonable cost and (b) facilitating foundry second-sourcing. The data points in Fig. 1.2 show that cell areas for digital process SRAMs have not changed nearly as rapidly and have reduced from 75 μm^2 in 1989 for a 1-poly, 2-metal BiCMOS process to 58 μm^2 in 1993 for an aggressive 2-poly, 5-metal BiCMOS process.

1.1.2 Circuit Design

In addition to improved processing technology, many innovations in circuit design have also driven SRAM performance. Some of these are the double-word line technique, data line equalization, hierarchical approaches to pipelining, the use of heavily pipelined self-resetting circuit blocks, and the development of current-mode sense amplifiers.

The last two of these have had the most pronounced effect on circuit performance. In [Cha91], researchers at IBM developed a 512kb CMOS SRAM with a fully pipelined, self-resetting architecture. A 4ns access time was achieved while using a modest 0.5μm 3-metal 6-

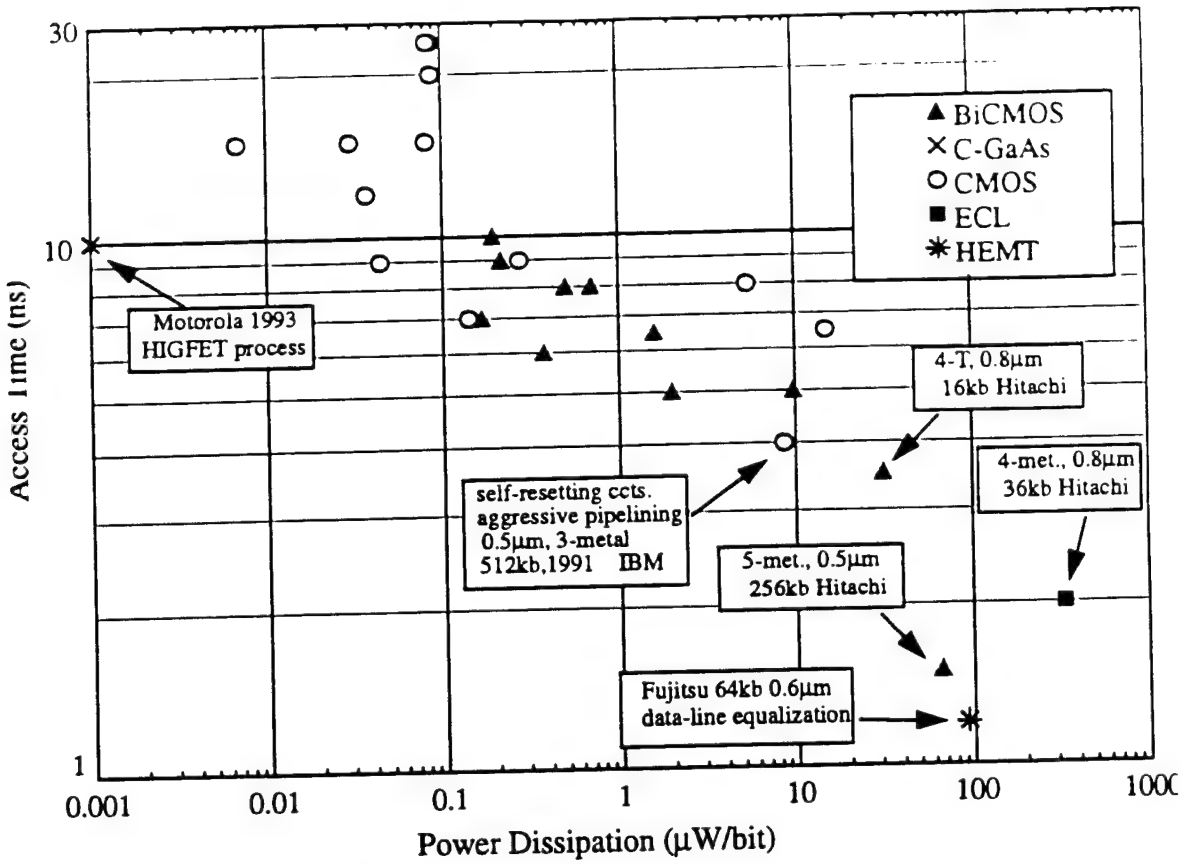


Fig. 1.3: Access time vs. per bit power dissipation
for CMOS, BiCMOS, ECL, HEMT and C-GaAs

transistor cell process. This represents the fastest speed for a large capacity CMOS RAM reported to date. Every circuit block in this RAM is a self-resetting circuit which is tuned to low-to-high transitions. Each circuit resets itself after a time adequate for its output to trigger the next stage. This is similar in principle to wave pipelining except that pulses rather than single transitions are propagated through the circuit. As shown in Fig. 1.3, this is the fastest CMOS SRAM to date (by about 50%) and has a 36 fJ/bit power-delay product. The drawback of using this technique for embedded applications is that the design cycle required to implement a RAM that uses this aggressive pipelining is very large.

The most significant circuit technique that has boosted CMOS, BiCMOS and ECL circuit performance has been the use of current-mode sensing techniques for readout, e.g. [Bla91],[Bla92], [Nam91]. Current-mode sense-amplifiers amplify a differential current on a pair of bit lines, rather than amplifying a differential voltage. There is a slight misnomer in the terminology used, however. All current mode sense amplifiers require a differential bit-line

voltage for reliable sense operation and thus have sensing speeds that are dependent on bit-line capacitance. This dependence, however, is much weaker than that of a conventional voltage-mode sense amplifier. Most sub-8ns SRAMs reported in CMOS and BiCMOS have made use of such techniques and operate with less than 20mV of bit-line separation

1.2 Circuit Design in Gallium Arsenide

There has been much speculation about the potential of using gallium arsenide (GaAs) for realizing high-speed VLSI circuits, with many arguments for and against using GaAs. Both the advantages and the disadvantages relate in some way to the properties of the materials, devices, and logic styles used to build GaAs circuits.

1.2.1 Why GaAs?

The most obvious advantage of GaAs over Si is that it has a higher electron mobility of 4000 compared to 800 $cm^2V^{-1}s^{-1}$ in Si at $N_D=10^{17}$ cm^{-3}. Ideally, this should lead to about a 5:1 increase in speed; practically, the performance of any system is degraded by parasitic resistances and capacitances. The real advantage in delay is closer to 2:1. As channel lengths become shorter, ballistic effects will cause much larger speed advantages.

Another desirable property of GaAs technology is that it has a semi-insulating substrate. This leads to lower parasitic on-chip capacitances.

The most mature GaAs device is the metal semiconductor field effect transistor (MESFET). MESFETs can be fabricated as enhancement-mode (E-mode) transistors that are normally off, or as depletion-mode (D-mode) transistors that are normally on. The most widely used logic style for GaAs MESFETs is enhancement/depletion direct coupled field-effect-transistor logic (E/D DCFL). This design style, which uses circuit structures that look similar to E/D nMOS circuits, achieves a comparatively low power dissipation and the highest density of the available design choices. DCFL circuits can operate at supply voltages as low as 1V without any degradation in performance. The facts that GaAs can achieve high speeds at lower electric fields than silicon, and that DCFL can operate on such a small supply voltage, suggest that GaAs circuits can be used to achieve faster circuits at lower power dissipations than is possible with silicon.

The fabrication of GaAs MESFETs requires fewer mask levels than are needed for competing technologies such as CMOS or BiCMOS. A conventional MESFET process flow requires only 4 masks to form the transistors, whereas advanced CMOS and BiCMOS process flows require about 10 to 16 masks just to define the transistors. Fewer masks lead to lower process complexity and lower processing costs. GaAs manufacturers believe that these simpler process flows will help to achieve devices with gate lengths of 0.25μm much more affordably than technologies requiring much more complex process flows, such as BiCMOS.

1.2.2 Why Not GaAs?

There are also many disadvantages to designing in GaAs. Many of these present circuit design challenges that must be overcome to achieve dense and reliable RAMs.

The gate of a MESFET is a Schottky diode. It therefore clamps the output voltage of a logic gate to one diode drop above ground when it is turned on. This results in a small signal swing and low noise margins. The smaller signal swings are a disadvantage because of the lower noise margins, but are also an advantage because they are responsible for much of the speed of DCFL.

DCFL is similar to ECL in that it is a current steering logic style. Consequently, DCFL circuits dissipate static power. This is a serious problem for circuits with low activity, such as a RAM.

Backgating is another problem that is related to the material properties of GaAs. A high density of energy states at the surface of the semiconductor creates a conductive path between adjacent devices. This coupling has an unpredictable impact on circuit performance, since currents flowing in one transistor are injected through the substrate into neighboring devices.

Large series source resistances in MESFETs precludes the stacking of transistors in series chains. This makes it difficult to implement NAND gates and pass transistor logic. Thus, the only feasible logic structures using E/D MESFETs are NOR and OR gates. Although boolean functions can be implemented using only NOR gates, this restriction increases the number of stages in a logic circuit that could otherwise take advantage of using alternative circuit structures.

Large process variations in GaAs limit the yield of large-scale integrated circuits using

E/D MESFETs. Although the control of threshold voltage is comparable to what can be achieved in CMOS, DCFL circuits are much more sensitive to process variations, since a ratioed logic style is used. Memory designs can also be seriously affected by these variations.

Other problems that present circuit design challenges in GaAs will be discussed in later chapters.

1.3 Thesis Overview

To leverage the benefits of a technology, it is necessary to develop design principles and circuit structures that are best suited for that technology. Until now, many of the circuit structures and design methodologies that are used in GaAs have been more appropriate for other technologies, such as CMOS.

As mentioned before, a key factor that will determine how well a high performance technology can compete will lie in the ability to integrate sufficient high speed memory on chip. The main focus of this thesis is research in novel GaAs static random access memories for embedded applications. The thesis presents several new circuit structures and design methodologies that are needed to achieve higher performance, lower power, process tolerant static RAMs and digital circuits using E/D MESFETs in GaAs.

Many technological problems have hindered the development of static memories in GaAs. In Chapter 2, we present a new current mirror memory cell (CMMC) that was invented to achieve higher performance memories to solve many of these problems. A 1-read, 1-write port asynchronous SRAM was designed as a vehicle to demonstrate this new approach to building RAMs in GaAs. Test results for the RAM are also presented.

The asynchronous SRAM that was designed to demonstrate the new memory cell suffered from low yield. This is due, in part, to circuits that were not designed to be tolerant of process variations in GaAs. In Chapter 3, we discuss yield considerations as they apply to GaAs circuit design. A new methodology is presented for the design and evaluation of circuits that are tolerant of process variations. As an example, a 4-read, 1-write port 2kb SRAM was designed using this methodology. Test results showed dramatic improvements in design yield for the memory, while also highlighting additional fault mechanisms in the CMMC. In this chapter, fault models for the

CMMC are presented along with test procedures appropriate for detecting these faults.

A problem with SRAMs and logic circuits made using E/D MESFETs, is that a large fraction of the power dissipated is consumed statically in gates where there is low circuit activity. In Chapter 4, we present a new logic style called power rail logic (PRL), that was developed to reduce the static power dissipation of specific circuit blocks within the RAM. This chapter presents ways that PRL can be used to reduce the power dissipation of some of the most common digital logic circuits, such as multiplexors, flip-flops, latches, and exclusive-OR gates. Chapter 4 also describes a new design methodology that was developed and incorporated into a tool to characterize and evaluate the merits of this new logic style.

The new memory cell described in Chapter 2, the process tolerant design techniques developed in Chapter 3, and the new logic style presented in Chapter 4, are used in the Aurora RAM compiler, presented in Chapter 5. This CAD tool represents a departure from the conventional methodologies used in RAM compilers. Because of the low noise margins and large process variations in GaAs circuits, this compiler iteratively optimizes the circuit for delay and noise margin using HSPICE as a simulation engine. The compiler was built using a flexible design framework that can be adapted with minimal effort to optimize and characterize memories in different MESFET processes. This feature was used to analyze the impact of processing technology on SRAM size, speed, and power dissipation.

Chapter 6 presents a conclusion and summarizes the contributions of this work.

CHAPTER II

THE CURRENT MIRROR MEMORY CELL

If GaAs is to impact the digital integrated circuits market, it must incorporate adequate amounts of high-speed memory on chip. High speed GaAs microprocessors developed to date [Upt93] have integrated only small amounts of memory on chip. Future designs will require large sub-2ns on-chip caches.

An overview of the history of GaAs MESFET SRAM development efforts, followed by technological problems that have hindered the progress of GaAs SRAM, is presented in Section 2.1. In Section 2.2, a new current mirror memory cell (CMMC) for GaAs MESFETs, developed to address many of these technological problems, is presented. The read and write operation of the CMMC are presented in Sections 2.3 and 2.4. The performance of memories using CMMCs is compared to the performance of memories using conventional memory cells. There are a few effects related to the CMMC that can prevent memories from functioning correctly. These fault mechanisms, which can be avoided by careful design, are identified in Section 2.5. Test procedures that detect the presence of these faults are also identified.

In Section 2.6, the test results for a 1k-bit SRAM that was designed to demonstrate the CMMC are presented. This design validated the new memory cell, but suffered from low yield. Using a methodology that addressed the yield problems identified in the asynchronous SRAM, a multi-port register file based on the CMMC was designed. Test results for the multi-port SRAM are presented in Section 2.7. Finally, concluding remarks are given in Section 2.8.

2.1 History of GaAs MESFET SRAM Development

The first GaAs SRAM was a 1kb E/D DCFL memory with 6ns access time and was

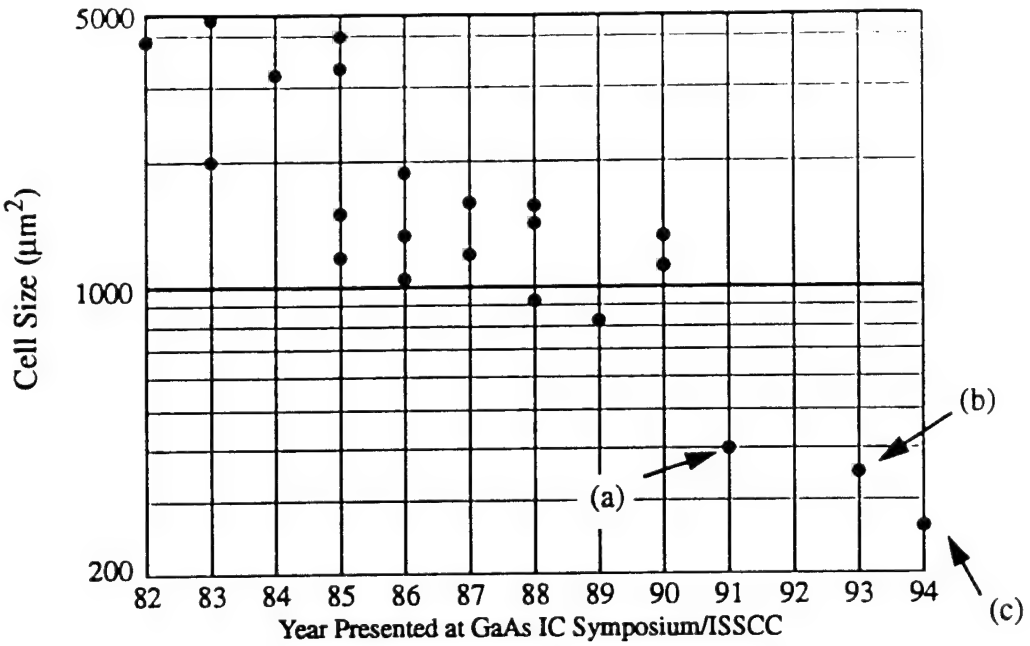


Fig. 2.1: GaAs MESFET SRAM cell area vs. year
(a) Vitesse (b) CMMC H-GaAs III (c) CMMC Motorola

reported by NTT in 1982[Ino82]. Since that development, more than 24 GaAs SRAMs, ranging in size from 1kb to 16kb have been reported. The progress in SRAM cell area is plotted in Fig. 2.1 as a function of time, in years. In the decade since their introduction, GaAs SRAMs have experienced a greater than 10-fold reduction in cell area. While the densities achieved are still a factor of 4 lower than those demonstrated in advanced Si processes, the gap is getting smaller. The smallest GaAs memory cell known has a cell area of 260μm^2. This cell was designed by the author in Motorola's Digital E/D MESFET GaAs process using the current mirror memory cell (to be discussed later).

GaAs SRAMs can be roughly categorized as follows:

- SRAMs made of D-mode only MESFETs using capacitor-diode FET logic (CDFL), e.g.[Fie86],[Fie88],[Fie90],[Tse87],
- SRAMs using D-mode MESFETs for access transistors, and high-speed peripheral circuitry, e.g.[Miz84],[Hir86],
- SRAMs using E/D DCFL with mostly-source-coupled FET logic (SCFL) peripheral circuitry e.g.[Tak85],
- SRAMs using mainly E/R and DCFL topologies e.g.[Gab87],

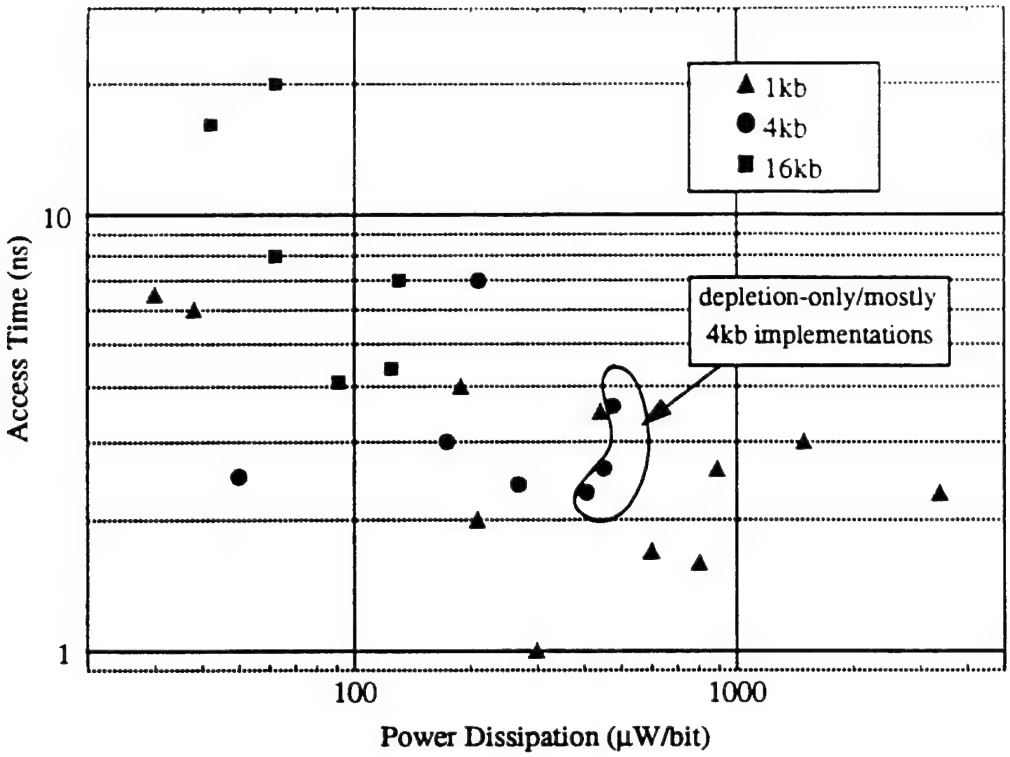


Fig. 2.2: Access time vs. per bit power dissipation for 1kb, 4kb, 16kb GaAs SRAMs

- SRAMs using mostly E/D DCFL e.g.[Hin91],[Hir84],[Mak90],[OCo85], [Tan86], [Ter88],[Toy86],[Nak90],[Tse87],
- SRAMs using JFET and C-JFET processes e.g.[Tro83], [Vog88].

The access times of reported GaAs SRAMs have been plotted in Fig. 2.2 as a function of the power dissipation per bit. Speeds as fast as 1ns for 1kb SRAMs have been achieved while 4kb SRAMs have demonstrated speeds between 2 and 3ns. For densities of 16kb, the fastest reported times have been just over 4.1ns[Hir86]. As densities increase, peripheral circuitry is amortized over a larger number of bits, and hence the power per bit figure decreases.

Consistently high yields have been demonstrated for many of the reported 1kb SRAMs. The power dissipations have varied dramatically from 30μW/bit to 4000μW/bit. This variation is not correlated to the logic style (DCFL, CDFL, CML, E/R etc.) used in the implementation. This is because 1kb SRAMs are typically prototyped to demonstrate a new process or logic style rather than to achieve the best power-delay product. Furthermore, for a size as small as 1kb, a large amount of the power is dissipated in the peripheral circuitry which supports only a small number of bits.

Consistently high yields have also been reported for 4kb GaAs MESFET SRAMs. The three implementations with the highest power dissipation were depletion-only and depletion-peripheral circuitry implementations. These dissipated 405-412μW/bit [Fie88], [Fie90], [Miz84]. The reason stated by authors for using mostly-D implementations was that much better access times could be achieved by not using circuit topologies with enhancement transistors. It is interesting that the E/D implementations were able to achieve equivalent delay, while dissipating only 50-270μW/bit.

The trend toward using E/D circuits becomes clear at integration densities of 16kb. Every reported 16kb implementation used only E/D DCFL. Access times as low as 4.1ns have been reported with power dissipations of 42 to 131μW/bit. Wafer-probe yields of 3 to 10% have been reported on 16kb parts.

Of the SRAMs reported, almost half require two or more power supplies and thus are not well suited for embedded applications. Of twenty-one reported GaAs SRAMs, nine required a single power supply, five required two supplies, five required three supplies, and one required four supplies. A second supply voltage is often used to power the memory cells, which require only 1V to operate. Of the remaining power supplies, some have been used to provide a large negative word line voltage to efficiently turn off memory cell access transistors, and others have been used to accommodate different logic styles on the same chip.

2.1.1 Significant Contributions in the Literature

Many of the SRAMs reported have been designed to demonstrate manufacturer process capabilities. There have been a few significant pieces of work that have drawn attention and have proposed solutions to specific challenges in GaAs. Some of these contributions are described below.

2.1.1.1 High Temperature Operation

Subthreshold leakage currents in MESFETs are orders of magnitude larger than those in MOSFETs. These currents consist of both gate-drain current caused by thermionic emission of carriers over the Schottky barrier and drain-source conduction. Measurements of the drain current

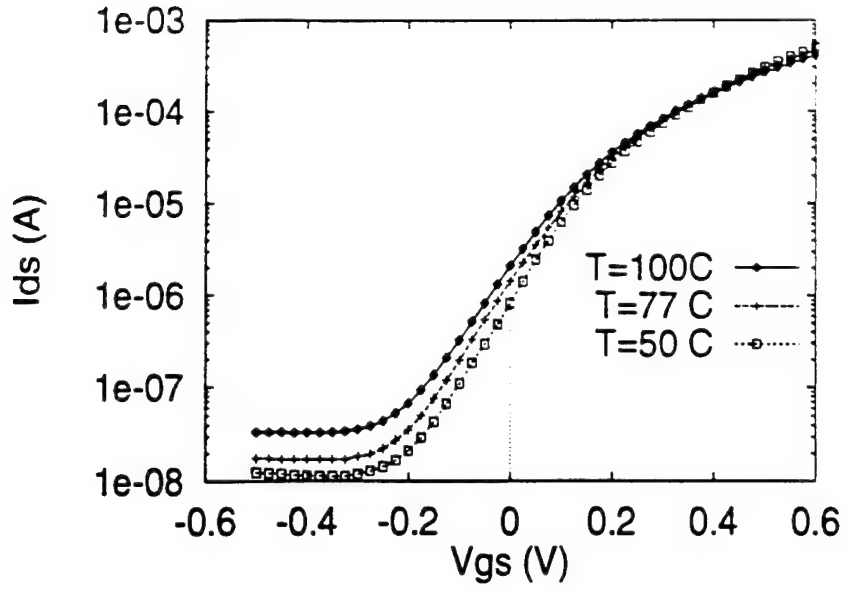


Fig. 2.3: Measurements of MESFET drain-source leakage currents as a function of temperature.

as a function of gate-source voltage for a 0.8μm long 20μm wide enhancement MESFET with V_{DS}=0.6V are shown in Fig. 2.3. At zero gate-source bias, the leakage currents can be as high as a few micro-Amperes.

As the number of bits per column is increased, the leakage currents associated with memory cell pass transistors that are turned "off" can corrupt the active current of a selected memory cell (see Fig. 2.4). This problem gets worse at higher temperatures. An approach to addressing this problem in the past was to include an extra negative power supply, -Vee, and to bring the word line voltage down to -Vee when turning a word line off. This is a very undesirable

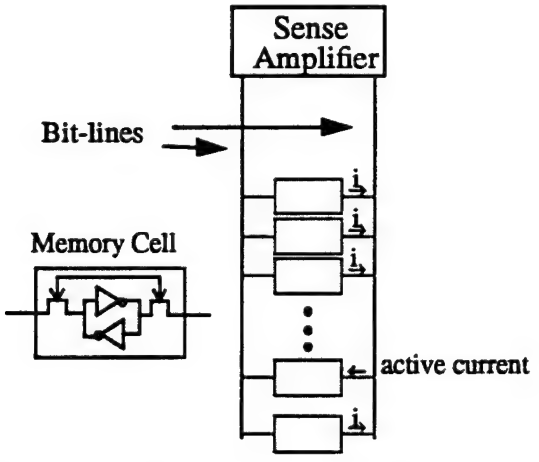


Fig. 2.4: The impact of leakage currents on the readout operation.

solution for embedded memory because it requires an additional supply voltage.

Makino et. al [Mak89] presented a solution that entails raising the ground level of a nonselected row to one diode drop above ground. By bringing the word line to ground and raising the cell ground, the access transistor gate-source junction becomes reverse biased. This reverse bias minimizes the leakage currents, as shown in Fig. 2.3. By reducing the leakage currents associated with nonselected rows, Makino et. al not only reduced the number of temperature-related failures, but also decreased the possibility of destructive readout.

2.1.1.2 Circuit Yield Limitations

Digital GaAs circuits did not start to use enhancement transistors until acceptable control of enhancement transistor threshold voltage, V_{TE}, was achieved. Even though the control of depletion transistors was worse than the control of enhancement transistors, logic families based on depletion-only transistors are tolerant of V_{TD} process variations. Hence, early digital GaAs circuits relied on depletion-only logic styles. When the σV_{TE} achieved levels below 40mV, E/D DCFL became much more viable and its use in digital logic circuits became more wide-spread.

A schematic of the conventional memory cell is shown in Fig. 2.5. M5 and M6 are depletion load transistors; M1 and M2 are enhancement switch, or access, transistors; M3 and M4 are enhancement driver transistors. Gabillard et. al [Gab87] performed statistical analyses to estimate memory circuit yield as a function of parametric process variations (also referred to as

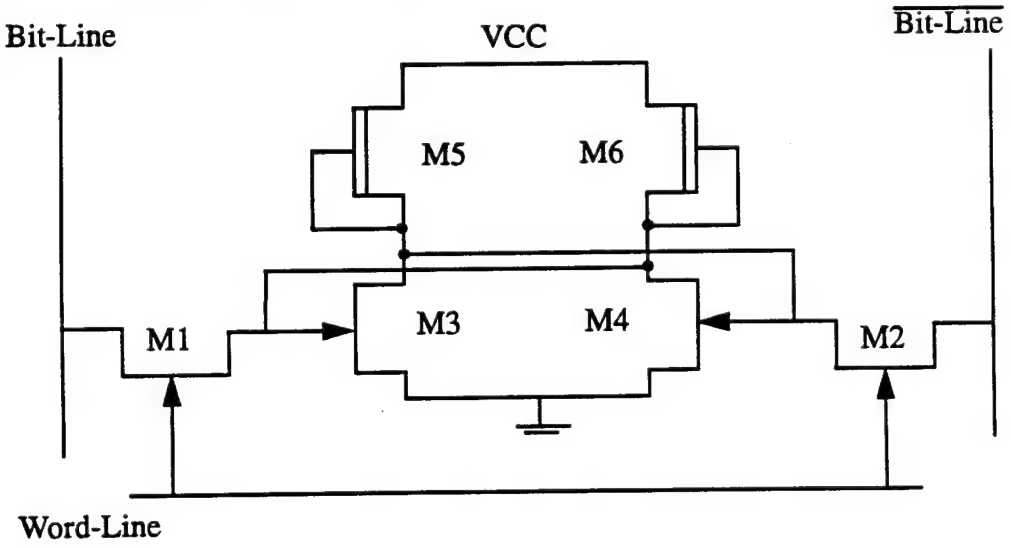


Fig. 2.5: Conventional memory cell

design yield). They found that the main factor degrading conventional memory cell design yields is local threshold variation between the driver and the switch transistors (ΔV_T). It is not enough to make the driver transistors in a memory cell three times as large as the switch transistors to prevent the possibility of destructive readout. They found that $\sigma(\Delta V_T)<10mV$ is required to achieve 100% design yields in the presence of these variations.

2.1.1.3 Access Time Scattering

Process variations cause a large scattering of access times across bits on a given chip. Access times for bits in the same memory array have been reported to vary from 2.5ns to 10ns by [Mak90]. Another report showed 4ns to 16ns variation with typical 7ns delays [Hay84].

Two different causes for this variation have been cited. The first authors to address this problem [Hay85] attributed it to V_{TE} scattering within the memory cell array. These authors presented bootstrap circuitry for the bit-lines to minimize this problem by uniformly making accesses faster. Greater than 50% reduction in access time scattering was reported by this group. Later authors attributed the variations to temperature-related leakage current effects [Mak90]. By introducing circuitry to reduce high temperature-related leakage currents, these authors achieved a 50% reduction in access time scattering.

2.1.1.4 Soft Error Rates

GaAs is more radiation hard to ionizing radiation than Si. Its radiation hardness is due to its higher bandgap of 1.4eV, compared to 1.1eV for Si. However, GaAs SRAMs are two orders of magnitude more susceptible to alpha particle induced errors than Si SRAMs. This is due, in part, to the much lower logic levels of 0.6V in E/D DCFL compared to 3.3V or 5V in CMOS SRAMs. The lower parasitic capacitances associated with memory cell storage nodes in GaAs memories also contributes to the higher soft-error rates.

To improve the immunity of the memory cell to alpha particles, one can either decrease the collected charge or increase the critical charge. CMOS has an advantage because the higher storage voltage leads to a higher critical charge. Two methods have been presented in GaAs to increase this charge. The first is to attach diode loads to the memory cell storage nodes to increase

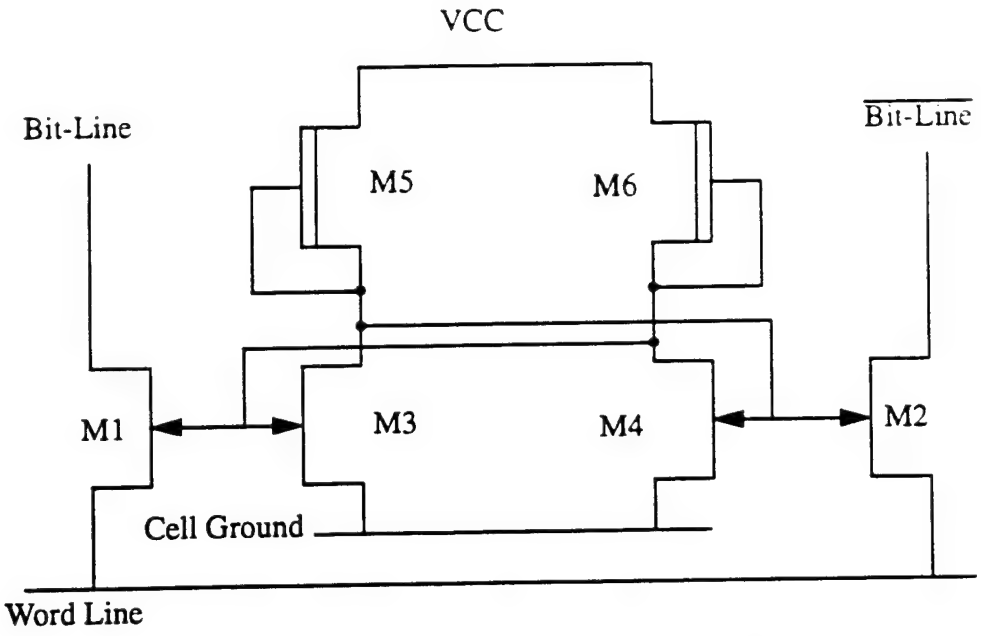


Fig. 2.6: Current mirror memory cell (CMMC).

the stored charge[Mat89]. The second method involves adding metal-insulator-metal capacitors to the storage nodes[Fie90]. Both methods have been shown to increase alpha-particle immunity by two orders of magnitude above that of commercial ECL SRAMs. The drawback of these methods is that the increased critical charge leads to slower write times.

2.2 The Current Mirror Memory Cell

To reduce the possibility of destructive readout, a conventional memory cell (Fig.2.5) utilizes driver transistors that are three times larger than the access transistors [Gab87]. To achieve high access speeds, however, large access transistors are necessary. These conflicting constraints lead to driver transistors in conventional cells that are much larger than minimum size, while the access transistors are smaller than desired. Fig. 2.6 shows the new current mirror memory cell (CMMC), designed to allow the driver transistors to be minimum size, while permitting the access transistor to be sized independently of the driver transistors. A cell of a given area and power is thus provided with a larger access current. The cell has greater immunity to destructive read problems than the conventional memory cell. In addition, it is not subjected to the read/write trade-offs associated with the conventional memory cell.

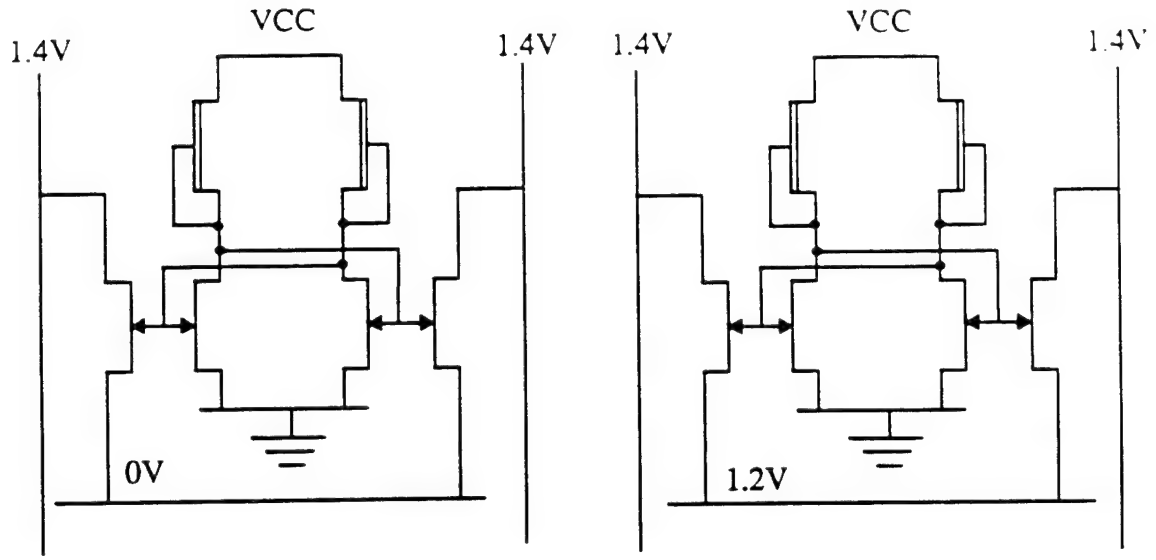


Fig. 2.7: CMMC cell bias.
(a) selected cell; (b) all other cells.

2.3 Read Operation

During the read operation, the various signal lines are biased as follows. Both bit-lines are initially at two diode drops above ground, or about 1.4V. The cell-ground line is a common ground signal for a given row. During the read operation, all cell-ground lines are biased at 0V. The word line for the selected row is lowered to 0V, while the remaining word lines are held at about 1.2V.

The access transistors of a selected memory cell, shown in Fig. 2.7a, are biased as current mirrors to the driver transistors. The gate-source voltages of the driver transistors and the attached access transistors are equal, while the drain-source voltages are different. The gate-source diodes (storage node to word-line) of the access transistors appear as an additional diode load to the memory cell. The gate-drain diodes (storage node to bit-lines) are reverse-biased and appear as additional capacitances to the storage nodes. Since the access transistors cannot inject current into the storage nodes, this read operation is non-destructive. As a result, the access transistors can be sized independently of the driver transistor without fear of destructive readout.

Previous efforts in GaAs have also used current mirror transistors for readout. These implementations have involved more complex memory cells with different write schemes [Lee93],[Fie86].

2.3.1 Leakage Currents

In a conventional GaAs memory, as the number of cells attached to a column is increased, leakage currents through nonselected access transistors can overwhelm the active current of the selected cell. Various methods have been proposed to reduce this problem for conventional memory cells [Mak90].

The biasing arrangement of the CMMC minimizes this leakage current problem. For simplicity the above description of the cell assumed that the cell ground and word line voltages were brought down to 0V. This would require impractically large pull-down transistors in the cell ground and word line drivers, which are designed to bring these signals down to about 0.1-0.2V. The memory cell storage nodes are at about 0.2V and 0.8V. The word lines remain at their standby bias of 1.2V. The bit-lines are initially at about 1.4V. As a result, both the gate-source and gate-drain diodes of the access transistors are reverse-biased by at least 0.4V. This arrangement minimizes the drain-source and drain-gate components of leakage current that flow onto the bit-lines. In the Vitesse process, this biasing arrangement allows a maximum of 512 memory cells to be safely connected to a column at 75°C, compared to a maximum of only 32 using conventional 6-transistor cells without a negative gate-source bias.

2.3.2 Non-Destructive Readout

Destructive readout has been identified as an important yield limiting factor for GaAs SRAMs[Gab87]. One advantage of the CMMC is the significantly reduced possibility of destroying the state of the cell during readout. In a conventional memory cell, when the word line is asserted high, two mechanisms cause noise to be injected into the cell. First, the low and high nodes of the cell become capacitively coupled to the bit-lines. Second, current is injected into the cell through the gate-source diode of the access transistor. This current injection is the more important mechanism and can make the state of the memory cell flip, causing destructive readout. As noted above, the driver transistors are typically made three times as wide as the access transistors to minimize the possibility of destructive read. The

When the word line of the CMMC is raised, charge is injected into both cell storage nodes through the non-linear gate-to-source capacitance. When the word line is lowered, charge is

removed from both storage nodes. This charge injection, which can cause problems in multiport RAMs, is described in greater detail in Section 3.4.

2.3.3 Access Current

The non-destructive readout properties of the CMMC also lead to advantages in read access current. The access currents of both the CMMC and a conventional memory cell are drain currents of the access transistors. This current is given by [Lon89]

$$I_{DS} = K \cdot (V_{GS} - V_{TH})^2 \cdot (1 + b(V_{GS} - V_{TH})^{-1}) \cdot (1 + \lambda V_{DS}) \cdot \tanh(\alpha V_{DS}) \qquad (2.1)$$

where K is the transistor transconductance parameter, V_{TH} is the threshold voltage, b is a velocity saturation parameter, λ is a channel length modulation parameter, and α is the drain voltage multiplier.

The CMMC is capable of providing much larger access transistor drain currents for faster readout than a conventional memory cell of the same area. This is because of two factors.

The first reason is related to the total area of the memory cell. In a conventional cell, an access transistor of width 2μm would require a 6μm wide driver transistor. A CMMC allows a 6μm wide access transistor while using only a 2μm wide driver transistor. The area of a well-laid-out memory cell is governed by transistor area, rather than by wiring. Thus, in the same area, one can use larger access transistors in a CMMC than in a conventional cell, thereby providing larger access currents.

The second reason is related to a limitation of the conventional memory cell. The following discussion assumes access transistors of equal size for both a conventional memory cell and a CMMC. The gate of the pass transistor in a conventional memory cell is biased by the word-line voltage (Fig. 2.5). This is usually clamped at one diode drop above ground to prevent destructive read operation. The source node of the pass transistor is connected to one of the storage nodes of the memory cell. As the pass transistor is made wider, more current is injected from the gate of the pass transistor into the cell. Both this noise current and the drain current of the access transistor cause the storage node voltages to shift, reducing the gate-source voltage applied to the access transistors.

The access currents are shown in Fig. 2.8 as functions of pass transistor width. The

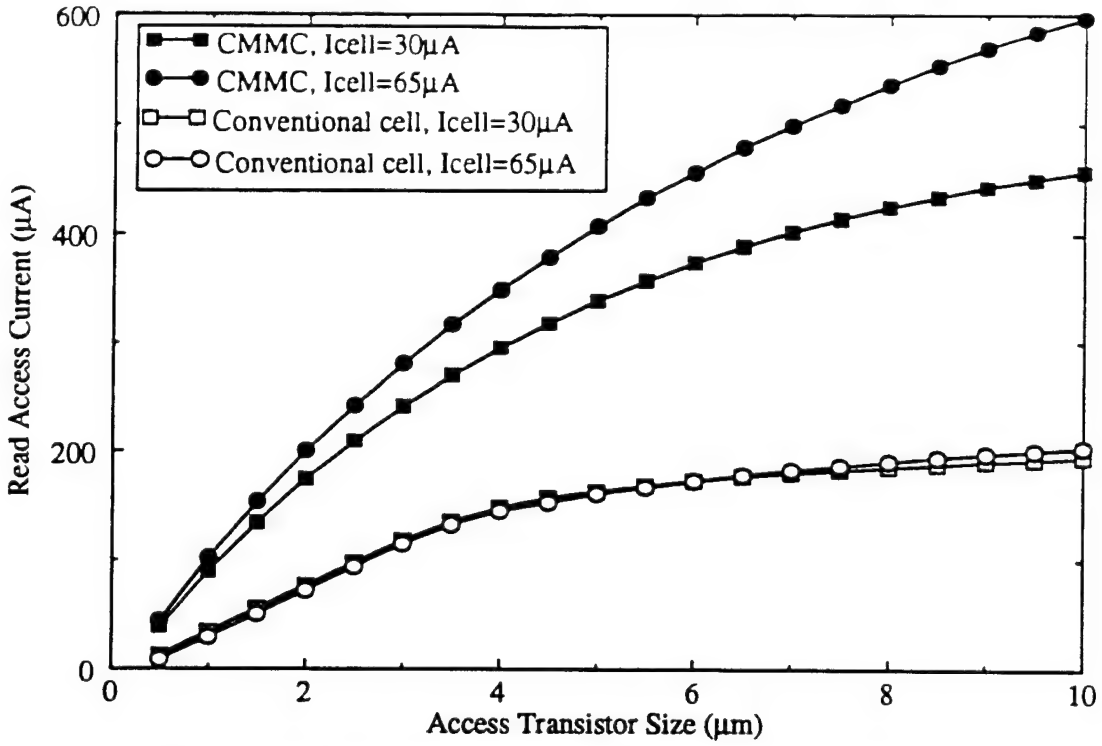


Fig. 2.8: Simulated access currents for conventional and CMMC.

memory cell driver transistors were scaled with the pass transistor size to maintain a three-to-one ratio. This graph shows that in conventional 6-transistor static RAM cells, when the width of the pass transistors exceeds 4μm, the access current starts saturating with respect to transistor width, rather than scaling linearly with it. This saturation is the result of increased current injection and charge sharing. In a CMMC, the gate of the access transistor is biased by the storage node voltage. This storage node is one diode drop above cell ground, which is close to circuit ground. The source node of this access transistor is biased by the word line, which is pulled low towards ground. As a result, the access transistor has a gate-to-source bias of

$$V_{GS}(access) = V_{CH} + V_{CG} - V_{WORD} \tag{2.2}$$

where V_{CH} is the cell high voltage (with respect to cell ground), V_{CG} is the cell ground voltage and V_{WORD} is the word line voltage at the cell.

Noise injection, which reduces the available access current in the conventional memory cell, is not a problem for this cell. The magnitude of access current can be further controlled by varying the cell ground bias with respect to the word-line bias, V_{CG} - V_{WORD}. This difference

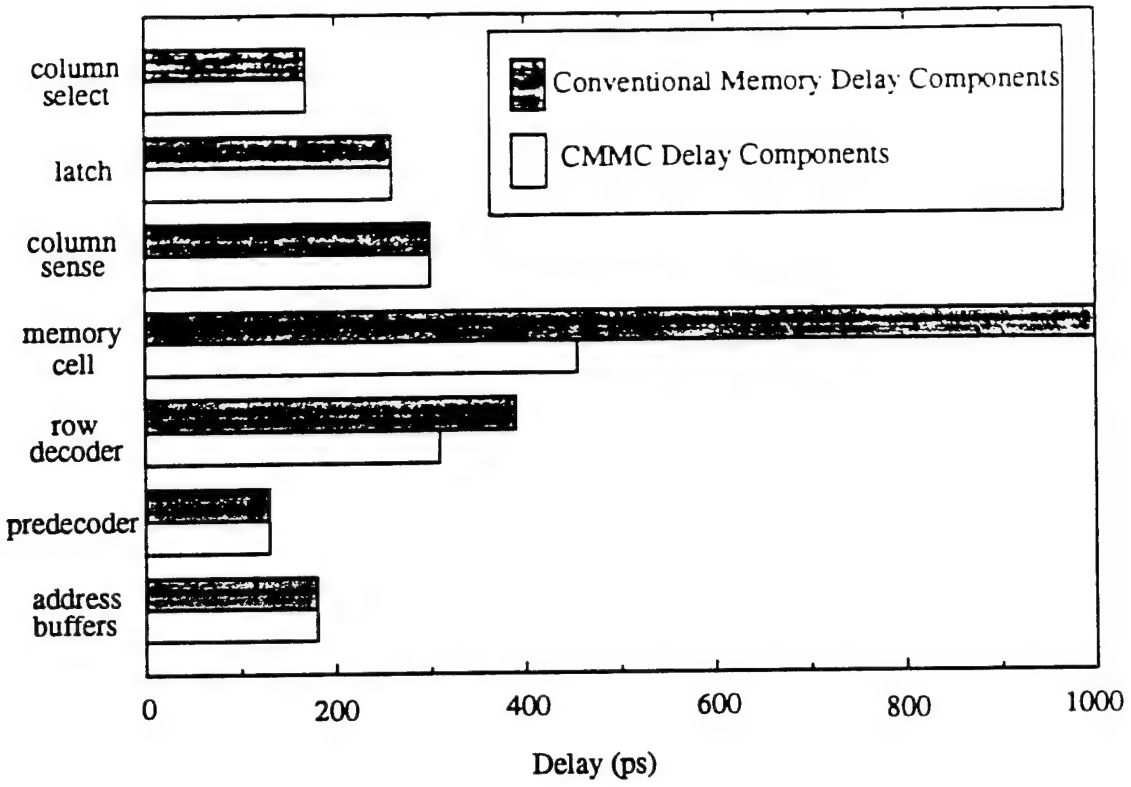


Fig. 2.9: Delay comparisons for the access times of a 4kb SRAM.

controls V_{GS} *(access)* linearly (2.2), which affects the access current quadratically.

The impact of this larger access current on access time is shown in Fig. 2.9, which compares through simulation, the delays of 4kb SRAMs of similar area, based on a conventional cell and a CMMC. For this size SRAM, the new memory cell offers a 25% decrease in the total memory delay. The impact is even more dramatic for larger SRAMs.

2.3.4 Adding Read Ports

Read ports can be added to a CMMC much more easily than to a conventional memory cell. For each additional read port, an extra word-line and pair of bit-lines is routed to the cell. Two additional access transistors are added to the cell, and are connected in the same manner as the access transistors associated with the first read port. Since the conventional memory cell is susceptible to destructive read problems, the cell current and driver transistor must be scaled when increasing the number of read ports. This scaling prevents destructive read if all read ports of a cell are accessed simultaneously. The driver transistor and load devices of a multiport CMMC do

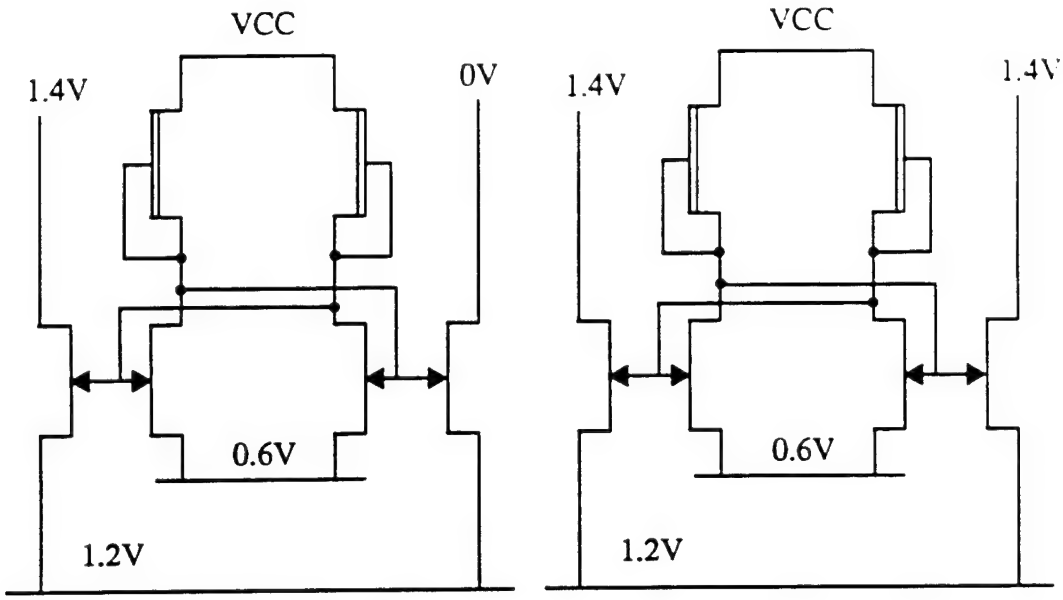


Fig. 2.10: CMMC cell bias during write.
(a) selected cell for writing; (b) other cells in selected row.

not have to be scaled nearly as much, however, which results in a smaller memory cell.

2.4 Write Operation

The technique used to write to the CMMC is very different from the method used to write to a conventional memory cell. In a conventional cell, the following sequence of events occurs. First, the row to be written is selected by asserting that row's word line. After a suitable delay, one of the bit lines is pulled low while the other remains high. The low-voltage bit line is connected through one of the pass transistors to a cell storage node, pulling that storage node low and causing the opposite cell storage node to be driven high.

The technique used to write to a CMMC is based on clamping the cell storage node low using diodes, rather than forcing the node low through a pass transistor. The biasing arrangement for a CMMC during the WRITE operation is shown in Fig. 2.10. A row is selected for writing by raising the cell ground line associated with that row above its normal system ground bias to one diode drop above ground, or about 0.6V. The word line remains at its standby bias of about 1.2V. A cell is written by forcing one of its two bit lines low while leaving the other one high. To understand how this arrangement writes to the cell, consider the equivalent circuit presented in

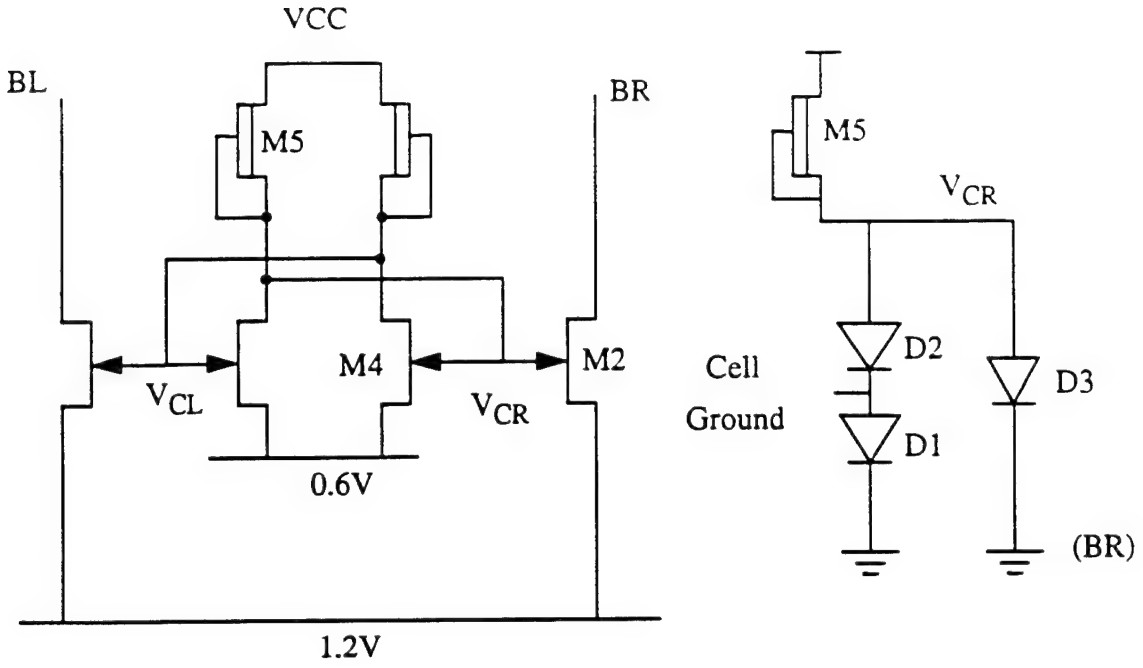


Fig. 2.11: Equivalent circuit of CMMC during write.
(a) memory cell bias; (b) equivalent circuit.

Fig. 2.11 One half of the cell is presented in Fig. 2.11(b).

After cell ground has been raised to one diode drop above ground, the cell storage nodes will be at 0.7V and 1.3V. Since the word line is biased at about 1.2V, the storage-node-to-word-line diodes of the access transistors will be off. The write action will take place through the storage-node-to-bit-line diode of the access transistor connected to the low bit line.

Transistor M5 is the depletion load feeding the gate of transistor M4. Diode D1 of Fig. 2.11(b) is a diode in the cell-ground driver that sets the cell-ground high-voltage to about 0.6V. Diode D2 represents the gate-source diode of the latch pull-down transistor M4. The gate-drain diode of M4 has been omitted. Diode D3 represents the gate-drain diode of access transistor M2. First, let us assume that before writing, V_{CR} was set to a logic high. When the cell ground is raised, V_{CR} would be at approximately 1.3V. When the bit-line is brought to ground, diode D3 turns on, clamping V_{CR} at one diode drop above ground, or about 0.6V, which is the same as cell ground. This causes a logic low to be stored on V_{CR}. The feedback of the latch causes V_{CL} to be set to a logic high. To store a logic high on V_{CR}, bit line BL is brought low while leaving BR high. This causes V_{CL} to be clamped low which sets V_{CR} high. The biasing arrangement for cells in the

selected row that are not being written to is shown in Fig. 2.10(b). Since the word lines remain at their high levels and neither of the access transistor gate-drain diodes is turned on, the cells maintain their states.

Cells in nonselected rows have their cell grounds held at 0V. If either of the bit lines is brought low, the storage node to bit-line diodes will appear as additional diode loads in parallel with the driver transistor gate-source diodes. Hence, it is impossible to write to a memory cell that is not in a selected row.

Fig. 2.12 is a timing diagram showing the state of the various control signals and cell storage nodes during a write and read operation. During the write operation, the read signal, shown in Fig. 2.12(a), is first lowered. This causes the cell ground line of a selected row to rise, as shown in Fig. 2.12(b), which in turn causes the memory cell storage nodes to rise. When one of the bit lines, shown in Fig. 2.12(a), is brought down to ground the memory cell storage node is clamped to cell-ground, causing the cell to be written. The write operation occurs here at 1.7ns.

Fig. 2.12(c) shows the word line. Initially, it is at a standby bias of approximately two diode drops above ground. During the write operation the word line becomes coupled through memory cell access transistors to the low-voltage bit-lines. The memory cell storage nodes prevent the word line from dropping any lower than one diode drop above ground, preventing any erroneous write through the word line. At 4.8ns in this simulation the word line is brought low for a read, causing the bit lines to separate.

The process of writing to the CMMC eliminates a race condition that complicates conventional memory design. Pipelined logic systems often specify that reads and writes occur on opposite phases of a system clock. A race condition occurs at the transition from a read to the following write. If the bit lines of a column are pulled low for a write before the word line from the previous read has been turned off, data may be erroneously written to cells at the previous address. This hazard is avoided by delaying the bit line signals long enough that the word line signal of the row with the slowest address reaches the memory cell first. This hazard avoidance can cost a significant fraction of the entire cycle time and make the write path the critical speed path [Hin91].

A CMMC can only be written if the cell ground line is raised to one diode drop above

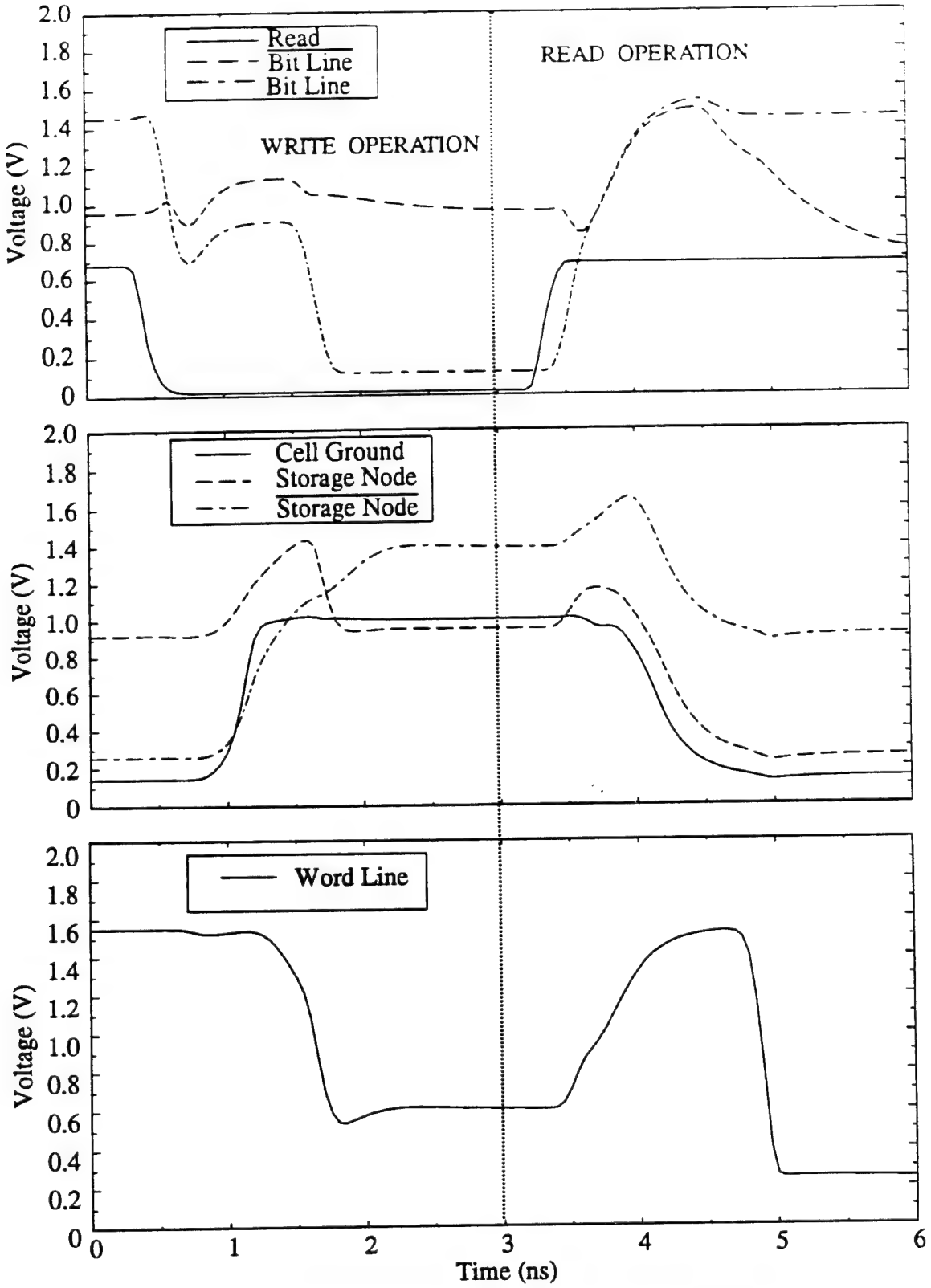


Fig. 2.12: Timing diagrams for the write and read operations.

ground. During a read, all cell grounds are at system ground. Thus, it is not possible to write erroneously to the cell at the address of the previous read. Since data cannot be mistakenly destroyed, there is no need to delay the bit line signals with respect to the row decode.

The write operation was described assuming that the cell ground was raised first, before the bit lines were pulled low. Whether the cell ground signal or the bit-line signal reaches the cell first does not matter. Since, in the CMMC, the bit line signals do not need to be delayed with respect to the word line or cell ground signals, simple asynchronous SRAM designs can be used in place of synchronous or self-timed designs.

The drawbacks of this memory cell are that cell grounds must be driven, and that additional write ports are expensive to add. The write signal for the cell is associated with the cell ground that is common to the entire row rather than a signal associated with a pair of access transistors. Thus, additional write ports would have to be added using pass transistors similar to those used in conventional memory cells. Fortunately, most applications for embedded memory require multiple read ports with only one write port.

Transferring data from the bit-lines to the cell usually occupies a significant portion of the total write time [Hin91]. The mechanism for writing to a conventional cell involves discharging the storage node through a pass transistor. The resistance of the channel of a transistor is a non-linear function of the drain-source voltage and is given in the region of operation (we are neglecting the $1 + \lambda V_{DS}$ term) by

$$r_{ds} = \frac{\partial V_{DS}}{\partial I_{DS}} = \left(\frac{\partial}{\partial V_{DS}}\left(K \cdot (V_{GS} - V_{TH})^2 \cdot (1 + b(V_{GS} - V_{TH})^{-1}) \cdot \tanh(\alpha V_{DS})\right)\right)^{-1}$$

$$= \frac{\cosh^2(\alpha V_{DS})}{K \cdot \alpha \cdot ((V_{GS} - V_{TH})^2 \cdot (1 + b(V_{GS} - V_{TH})^{-1}))} \qquad (2.3)$$

As a cell is being written, V_{DS} of the access transistor is reduced, resulting in a lower channel resistance. The method used to write to the CMMC involves discharging the storage node through a Schottky diode, which has a resistance given by

$$r_{gs} = \frac{\partial V_{GS}}{\partial I_{GS}} = \left(\frac{\partial}{\partial V_{GS}} I_O \cdot \left(exp\left(\frac{qV_{GS}}{nkT}\right) - 1\right)\right)^{-1}$$

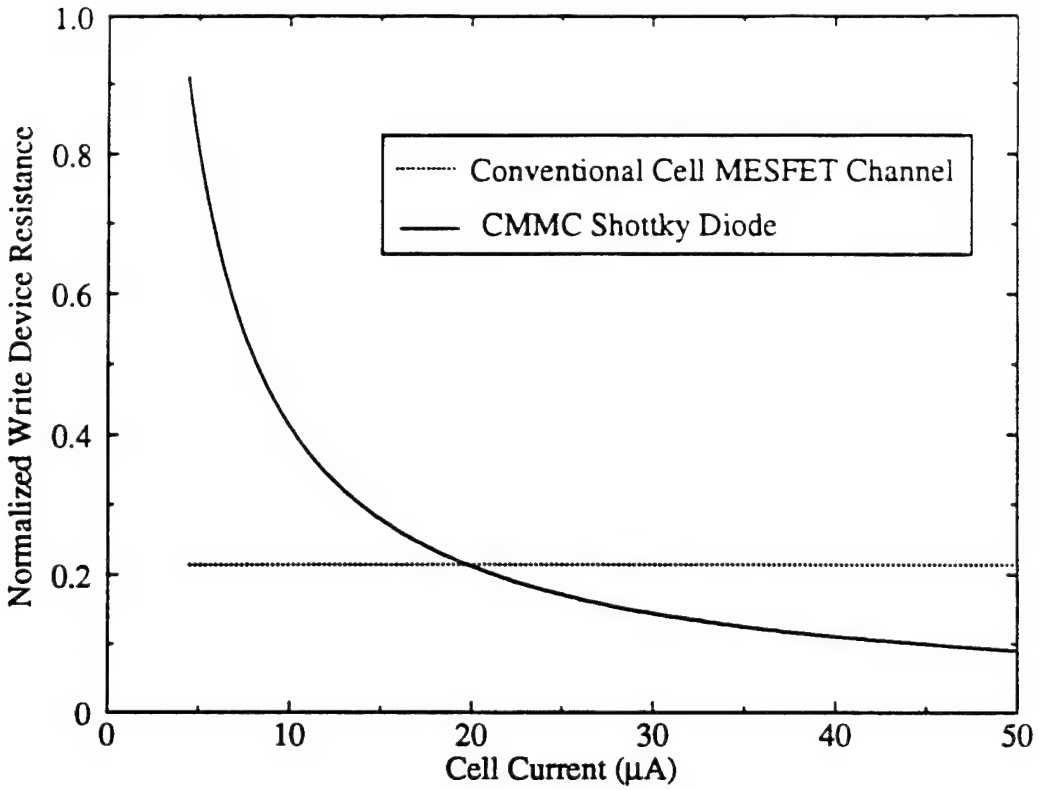


Fig. 2.13: Comparison of Schottky diode and MESFET channel resistance.

$$= \frac{nkT}{qI_{GS}} \approx \frac{2nkT}{qI_{CELL}} \tag{2.4}$$

The resistance of the Schottky access transistor diode is inversely proportional to the cell current. A normalized comparison of the on-resistance of the CMMC Schottky write diode and the on-resistance of a MESFET channel averaged over the write time is shown as a function of cell current in Fig. 2.13. This simulation was performed using 4μm wide access transistors for both cells. For cell currents above 20μA, the CMMC exhibits smaller write-device on-resistance and thus faster write times. Below this level, conventional cells can achieve faster write times. For cell currents below 10μA, leakage currents within a cell become much more important and can upset the state of the cell.

2.5 CMMC Fault Mechanisms and Test Procedures

There are many sources of failures in SRAMs, including component density, circuit lay-out, circuit design, and method of manufacture. Fault models describe the manifestation of these

faults during circuit operation. In this section, four mechanisms that can cause failures in memories designed using the CMMC are described. Each of these failure modes can be avoided by careful design.

2.5.1 Leakage Current Faults

As discussed in Section 2.1.1, subthreshold leakage currents can cause readout problems even in the CMMC design. As the number of bits per column is increased, the leakage currents associated with access transistors that are turned "off" can corrupt the active current of a selected memory cell, as shown in Fig. 2.4. Although the biasing scheme for the CMMC is intended to minimize the impact of these leakage currents, a design that either is intolerant of process variations or has a poor power distribution network may result in leakage current problems.

This fault can be detected using any test procedure that fills a column with all 0s (or 1s), and then writes a 1 (or a 0) to any bit within that column successfully. The fault would be detected if the memory cell storing a 1 (or 0) could not be read properly.

2.5.2 Bit-Line Coupling Fault

A second fault that is specific to the CMMC occurs when the data stored in memory cells in a column affect the signal swing of the bit-lines during a write. This can be visualized with the aid of Fig. 2.14. When a bit-line is lowered towards ground, the memory cell access transistors' storage-node to bit-line junction biases increase. The high-voltage (H) storage nodes will start to couple to their associated bit-lines before the low-voltage storage node. If the values stored in a column are all identical, as shown in Fig. 2.14, then there is a strong tendency for the word-line high voltages to couple to the bit line associated with the high-voltage storage nodes, preventing the bit line from dropping lower than one diode drop above the cell-ground low voltage. Therefore, the cell-ground must be raised by more than one diode drop above ground to prevent this bit-line clamping from interfering with the write process. In more recent designs, the cell-ground voltage is typically raised to about 1V during a write operation (as shown in Fig. 2.12b).

The algorithm described above for testing the leakage-current fault also tests for the bit-line fault. By filling the memory array with all 1s (or 0s) and writing a 0 (or a 1) to a single cell,

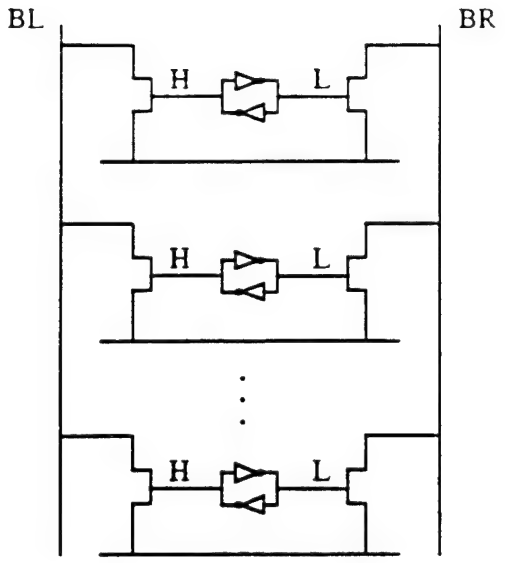


Fig. 2.14: Data-dependent bit-line clamping.

In this column, all the storage nodes associated with BL are high (H). When BL is lowered, the storage nodes limit the bit-line swing.

the ability of the memory to write to a cell in spite of the influence of other cells in the same column is tested.

2.5.3 Word-Line Resistive Voltage Drops

A third type of fault is related to resistive voltage drops along the word line. During the read operation, the gate-source bias of memory cell access transistors is given by

$$V_{GS}(access) = V_{CH} + V_{CG} - V_{WORD}$$

where V_{CH} is the cell high voltage (with respect to cell ground), V_{CG} is the cell ground voltage and V_{WORD} is the word line voltage at the cell. Both the cell-ground line and the word-line exhibit significant distributed I-R drops over the length of their wires, as illustrated in Fig. 2.15. The word line in a row is brought low for the read operation. The read-access current is typically 300μA whereas the cell current is usually below 60μA. Hence, the resistive drop along the word line is much larger than the drop along the cell-ground line. The word line will have its largest value at the memory cell furthest away from the word-line driver, leading to a smaller access-transistor gate-source voltage. A 50mV reduction in access transistor gate-source voltage can lead to a

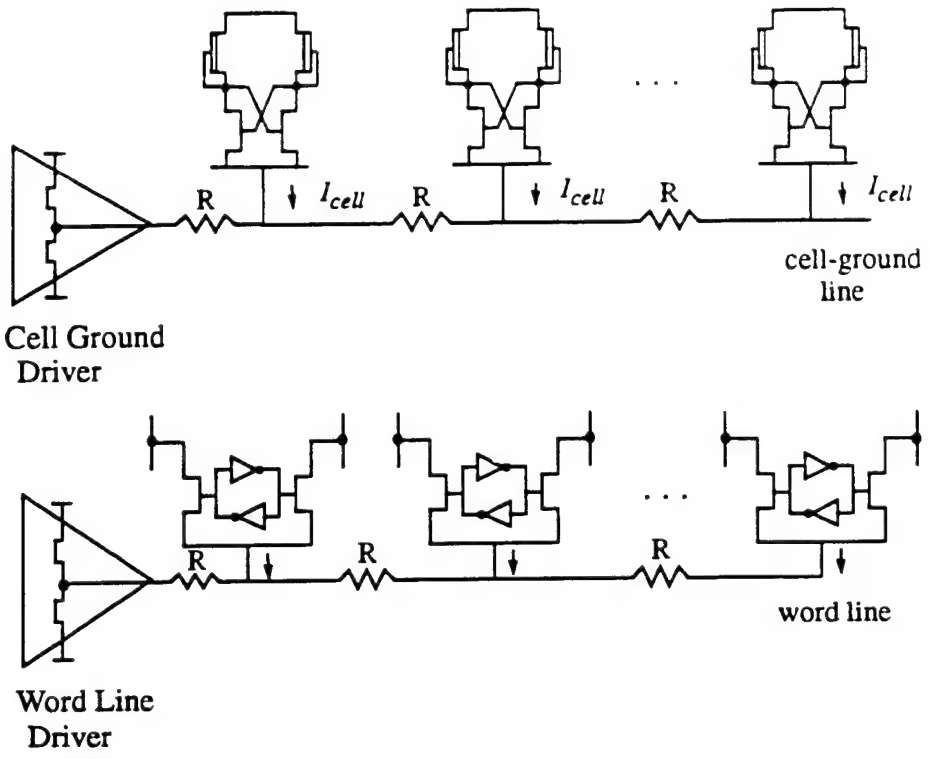


Fig. 2.15: Resistive drops along the cell-ground line and word line.

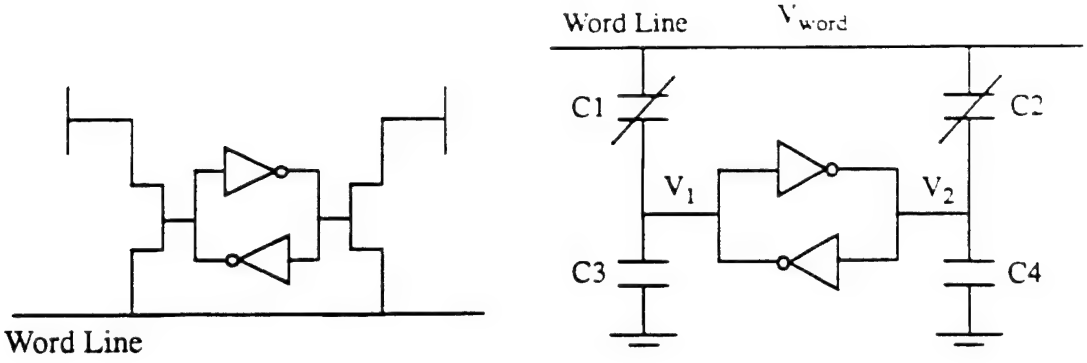
reduction in read-access current of 50%, thus seriously degrading the read time of the memory cell. If the voltage drop along this line is large enough, it can result in read-access currents that are too low for functional read operation. Therefore, any test procedure that tests for stuck-at bits will detect this error.

2.5.4 Word-Line Induced Charge Injection

A fourth fault associated with the CMMC is word-line induced charge injection. When the word line is either lowered or raised, charge is injected into the memory cell through the gate-source capacitances of the access transistors, as shown in Fig. 2.16. The voltage-dependent effects of the gate-source capacitance are removed by using an equivalent linear capacitance to calculate the charge injected into the memory cell. For a voltage transition across the capacitor between biases V1 and V2, where V2>V1, the equivalent capacitance is given by [Hod83]

$$C_{equiv} = \frac{\Delta Q}{\Delta V} = \frac{Q(V_2) - Q(V_1)}{V_2 - V_1} = K_{eq} C_{jo} \tag{2.5}$$

where



(a) (b)

C1, C2: Access transistors' gate-source capacitances.
C3, C4: Memory cell storage node capacitances.

Fig. 2.16: Word-line induced charge injection into the current mirror memory cell.

$$K_{eq} = \frac{-2\phi_o^{0.5}}{V_2 - V_1}\left[(\phi_o - V_2)^{0.5} - (\phi_o - V_1)^{0.5}\right] \tag{2.6}$$

and

$$C_{jo} = \sqrt{\frac{q\varepsilon_{GaAs}N_D}{2\phi_o}} \tag{2.7}$$

When the word line is raised or lowered, charge is injected or removed, through capacitors C1 and C2, from both storage nodes simultaneously. Since the storage node voltages are different, so are the gate-source voltages of the access transistor. This results in different values of C1 and C2, causing an asymmetrical amount of charge to be transferred into or out of the two storage nodes.

The change in memory cell storage node voltage is given by

$$\delta V_{storage} = \Delta V_{word} \cdot \frac{C_{equiv}}{C_{equiv} + C_{storage}} \tag{2.8}$$

where ΔV_{word} is the change in word-line voltage, C_{equiv} is the capacitance defined in (2.5), and $C_{storage}$ is the storage node capacitance.

Calculations show that for an access transistor to driver transistor ratio of 3:1, and a

Step 1. Write: C_i <- 0 for i=0, 1, ..., n-1
Step 2. For i=0, 1, ..., n-1
Read: C_i (=0)
Write: C_i <- 1
For all $j \neq i$
Read: C_j (=0)
Read: C_i (=1)
Write: C_i <- 0
Step 3. Repeat Steps 1 and 2, interchanging 1's and 0's.

Fig. 2.17: The modified GALPAT test procedure.

storage node capacitance of 20fF, the difference between the noise injected onto the two storage nodes is only about 20mV. While this amount of noise can not corrupt the state of a memory cell that has only one read and one write port, it can cause destructive read in a multiport SRAM if not properly designed.

2.5.5 Test Procedure Selection

In addition to the faults described here, generic fault models for semiconductor memories also apply to memories designed using the CMMC. These modes include stuck-at-0/1 faults, coupled-cell faults, and decoder faults. The modified galloping pattern (GALPAT), Thatte-Abraham, and Nair-Thatte-Abraham test procedures each cover all decoder, stuck-at-0/1, and coupled-cell faults[Abi83]. The number of test vectors required for GALPAT is $O(n^2)$, for Thatte-Abraham is $O(n\log n)$, and for Nair-Thatte-Abraham is $O(30n)$, where n is the number of bits in the memory array.

Of these 3 procedures, GALPAT is the only one that fully tests for leakage current and clamped bit-line faults. The GALPAT algorithm is described in Fig. 2.17. In this description, the notation C_i <- d is interpreted as write data d into cell i.

2.6 Demonstration Vehicle

An asynchronous 1kb SRAM was designed to develop and demonstrate the CMMC. The

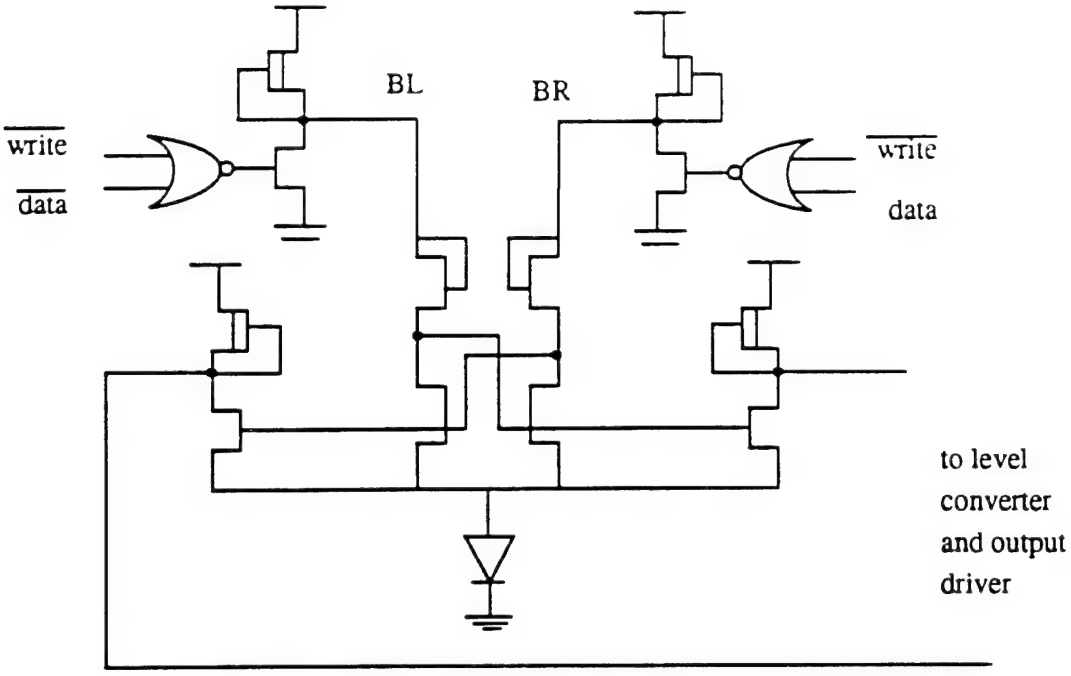


Fig. 2.19: Sense amplifier and write circuitry.

chip was fabricated through MOSIS in the Vitesse H-GaAs III, 4-metal, 0.6μm E/D MESFET technology. Fig. 2.18 is a photomicrograph of this RAM.

The primary reason for making the design asynchronous was to satisfy the small-memory needs of the Aurora microprocessor system. The asynchronous design can provide sub-2ns access times for design sizes to 4kb. The relaxed timing constraints of the CMMC facilitates rapid prototyping of RAMs of different configurations and even different process technologies, without concern about read after write hazards or the need to generate self-timed pulses.

The column circuitry of this SRAM, including sense amplifiers and write drivers, is shown in Fig. 2.19. To minimize the impact of process variations on the functionality of the circuit, the sense amplifier was made primarily of enhancement transistors. The cell ground and word line drivers used in this SRAM are shown in Fig. 2.20. The two diodes have been omitted in implementations described later in this thesis, thus achieving identical performance at lower power dissipation. In memories we have implemented which have a larger number of bits per column, the cell ground driver is designed to raise the cell ground line to about 1.0V instead of just 0.6V to ensure adequate write margins. Most of the high capacitance drivers in the predecode logic and signal buffers in the SRAM are made from enhancement-type superbuffers.

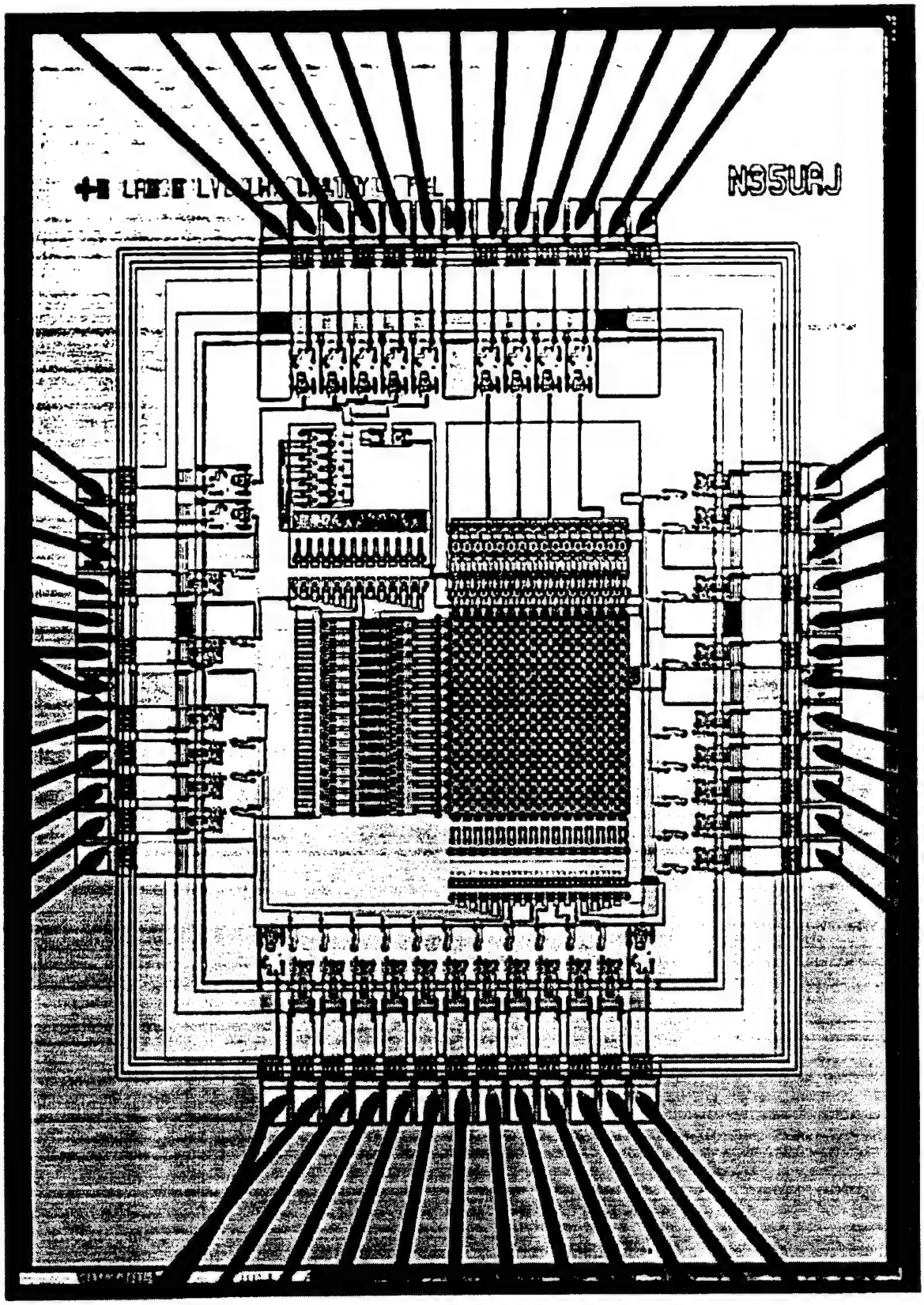


Fig. 2.18: Photomicrograph of the 1kb SRAM demonstration vehicle.

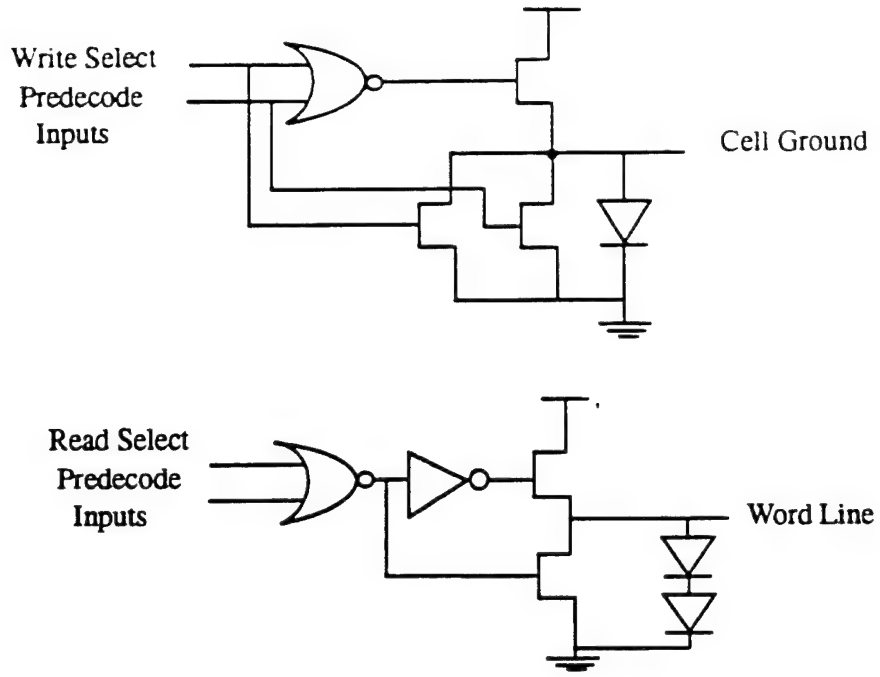


Fig. 2.20: Cell-ground and word-line drivers.

Twenty chips were packaged, and tested on a Hewlett Packard 82000 IC Evaluation System. Three different functional tests were run to identify different types of faults. In the first test, all memory locations were first written with ones. All locations were then read. This procedure was then repeated by writing and then reading all zeros. The purpose of this test was to expose any stuck columns. The second test was a checkerboard pattern. The purpose of this test was to uncover any slow columns or stuck bits that were not found in the first test. In the third test, the entire memory array was filled with zeros. A single row was subsequently written with all ones. All rows and columns were then read. The purpose of this test was to determine whether the leakage currents of all rows filled with zeros were enough to prevent correct writing or reading of the row to be filled with ones. Of the twenty chips, two were non-functional. Six chips passed the first test, four passed both the first and second test and three passed all three tests, resulting in 15% packaged yield. The pattern of bit failures in chips that did not pass any of the three functional tests can be broken down into three categories of errors. Roughly one third of the errors noticed were random stuck bits. One quarter of the errors were collections of stuck bits within a column. The third cause of yield problems was determined to be inadequate power distribution to the output pads. Most of this type of error disappeared when the tests were set up so a maximum of

Table 2.1: Summary of 1kb SRAM characteristics.

Parameter	Value
Organization	32 x 32
I/O Levels	GaAs/ECL
Supply Voltage	2V
Cell Size	$350\mu m^2$
Cell Current	60μA
Chip Size	1.3mm x 1.1mm
Address Access Time	1.0ns - 2.3ns
Write Time	1.0ns
Power Consumption	800 mW

eight outputs were switching simultaneously.

The test results are summarized in Table 2.1. The power budget for this chip is categorized in Table 2.2. Dramatic improvements in power dissipation could be achieved by eliminating the two clamping diodes in the word line drivers, using enhancement-type push-pull drivers for the write circuitry, and using a special weaker depletion implant for the memory cell load transistors. Some of these improvements have been used in other implementations described in this thesis.

Address access times were measured over 512 bits of the fully functional chips. Histograms of the access times for reading a one and a zero for one of these chips are given in Fig. 2.21. These on-chip delays were obtained by measuring the raw delays and subtracting measured

Table 2.2: Power budget for 1kb SRAM.

Circuit Block	Simulated Power Dissipation
Input Buffers	32 mW
Predecode Circuitry	100 mW
Row Drivers	266 mW
Write Circuitry	200 mW
Cells	288 mW
Sense Amps	60 mW
Total	946 mW

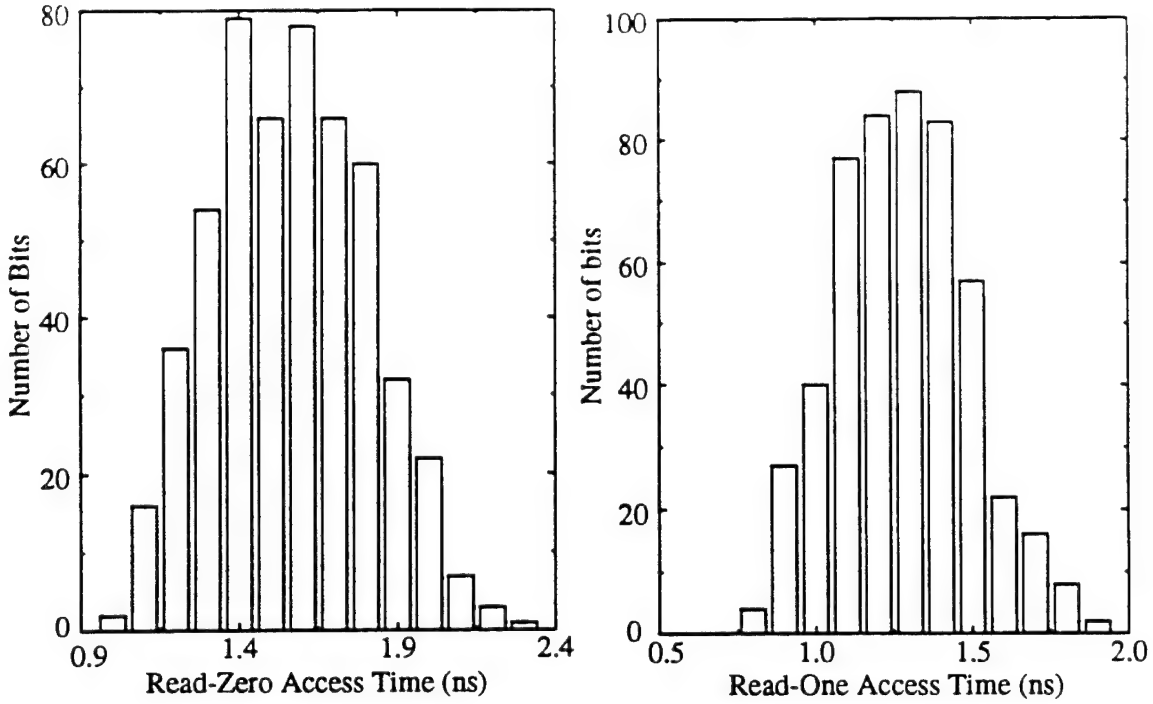


Fig. 2.21: Measured read access time scattering for 1kb SRAM.

input- to output-pad delays. The measured data show bit access time scattering from 1.0ns to 2.3ns, with an average address access time of 1.56ns and standard deviation 250ps. The HSPICE simulated access time was 1.2ns using typical parameters. The threshold voltages of transistors on the wafer from which the chips came were much higher than expected and hence explain the slower average access times.

Most of the access time variation was observed between columns rather than within columns. The source of this variation is probably process variations in the sense amplifier. A write pulse of width 1.0ns was adequate to write successfully to all bits. An oscilloscope trace taken from this chip is shown in Fig. 2.22. This figure shows (below) an input address transition and the resulting (above) complementary output data transitions. For this bit, the off chip access time is 2.4ns. With 1ns measured pad delays, this results in a 1.4ns on-chip access time. These measurements were all taken at room temperature. Junction temperatures on the chip were estimated to be 80°C. No correlation between the number of failures and the chip temperature were found[Pu94].

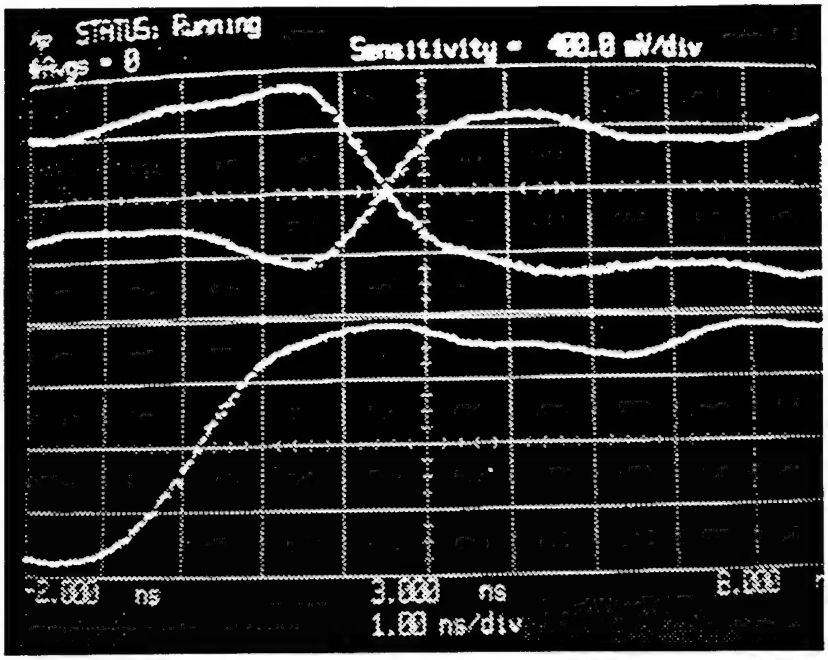


Fig. 2.22: Typical data output and address-input waveforms.

2.7 A 5-port Register File

A synchronous 5-port register file was designed and fabricated through MOSIS in the Vitesse H-GaAs III, 4-metal, 0.6μm E/D MESFET process. Fig. 2.23 is a photomicrograph of this SRAM. The test chip consists of the register file and logic to increase the controllability and observability of the memory.

This synchronous SRAM was designed to satisfy the multi-port register file need for a floating point unit[Huf95]. A major source of failure in the asynchronous 1kb SRAM previously discussed was stuck columns. In Chapter 3, a design methodology is presented for optimizing the process tolerance of sense-amplifiers. This methodology was applied to the sense-amplifier used in this floating point register file (FPRF). This SRAM was also built to test some novel circuit structures, and to determine whether large memories with acceptable yields can be built in GaAs. In this section, some of the features of this SRAM, chip testing results, and an analysis the observed failure modes are described.

Some of the chip's features are summarized in Table 2.3. The SRAM is organized in 32 rows by 64 columns. To minimize the I-R drops along signal lines, the array is divided in two as shown in Fig. 2.23. The memory has 4 read ports and 1 write port. The size of the die is 7.4mm x

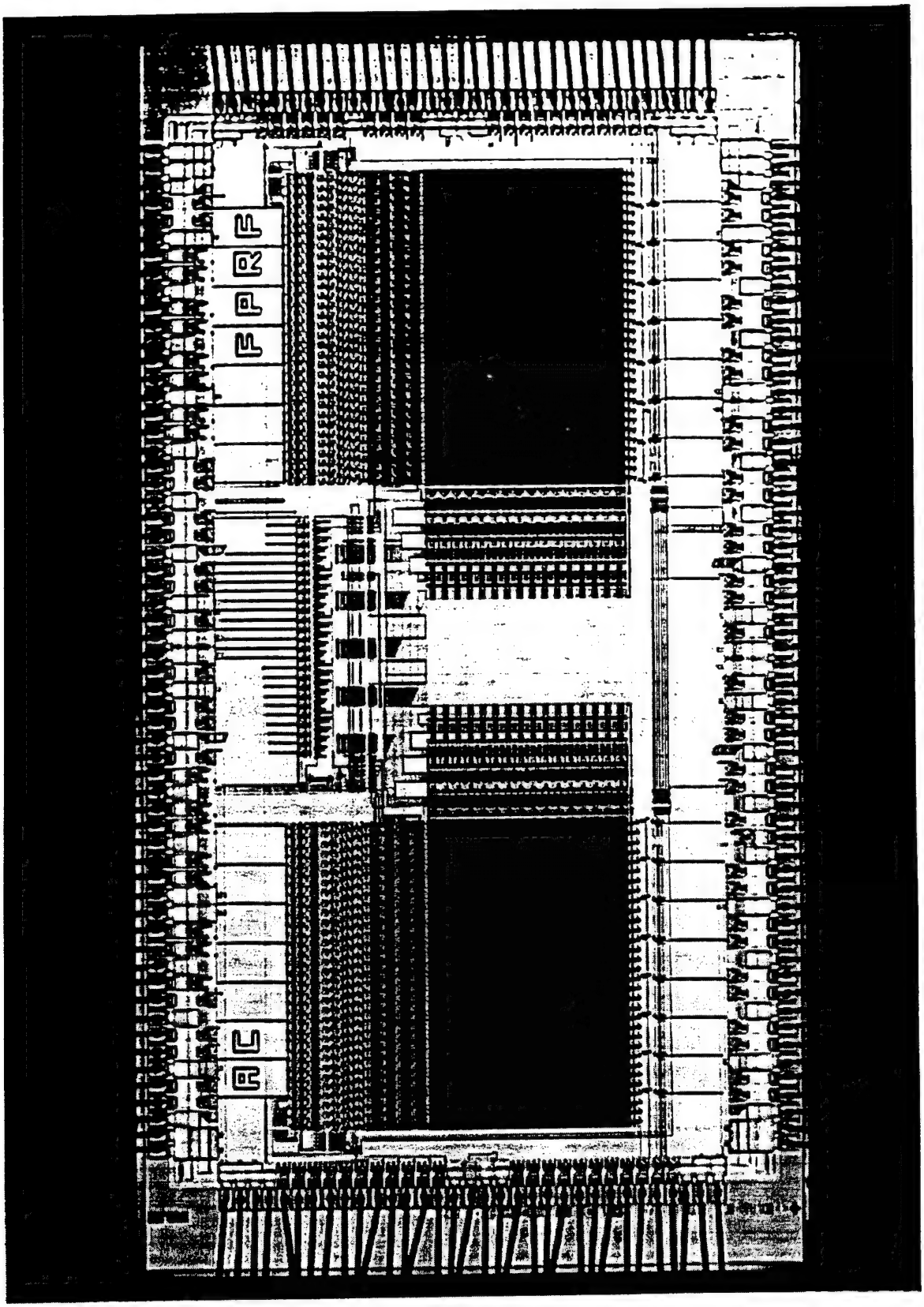


Fig. 2.23: Photomicrograph of the 5-port synchronous SRAM.

Parameter	Value
Organization	32x64
Read Ports	4
Write Ports	1
Sense Amps per chip	256
Cell Size	$2352\alpha\mu m^2$
Die Size	7.4mm x 4.8mm
I/O Levels	ECL, GaAs, GTL
Package	149-pin PGA

Table 2.3: Floating point unit register file test-chip characteristics.

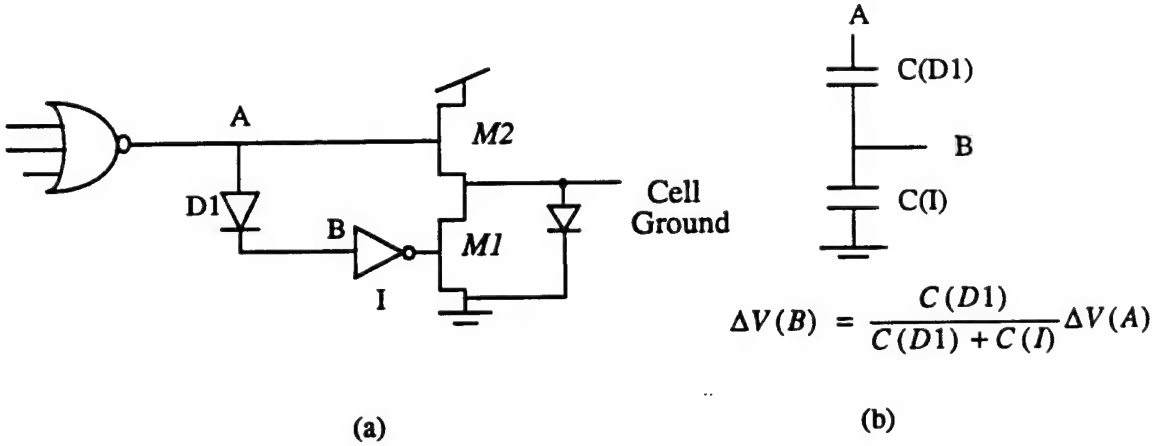


Fig. 2.24: Logic level splitting scheme.
(a) Cell-ground driver used for the floating-point register file;
(b) determination of low-voltage level for signal B.

4.8mm, or $35.5mm^2$, which is large enough to adequately indicate the yield of larger circuits.

The read and write ports are accessed synchronously by a clock that is used to generate an internal self-timed pulse. A synchronous design was chosen over the asynchronous design presented in the previous section to allow larger bit-line swings for increased process tolerance without degraded the memory performance. The synchronous design also required different write circuitry and additional synchronizing circuitry. Details of the support circuitry are presented in Chapter 5.

The FPRF employs a novel scheme to derive different logic levels from the same wire using a diode. Fig. 2.24 shows how this technique is used in a cell-ground driver. In this circuit, diode D1 allows both a buffered-FET logic (BFL) signal and a DCFL signal to be derived from node A. Since the input-gate to inverter I appears as a diode to ground, the high-levels for nodes B

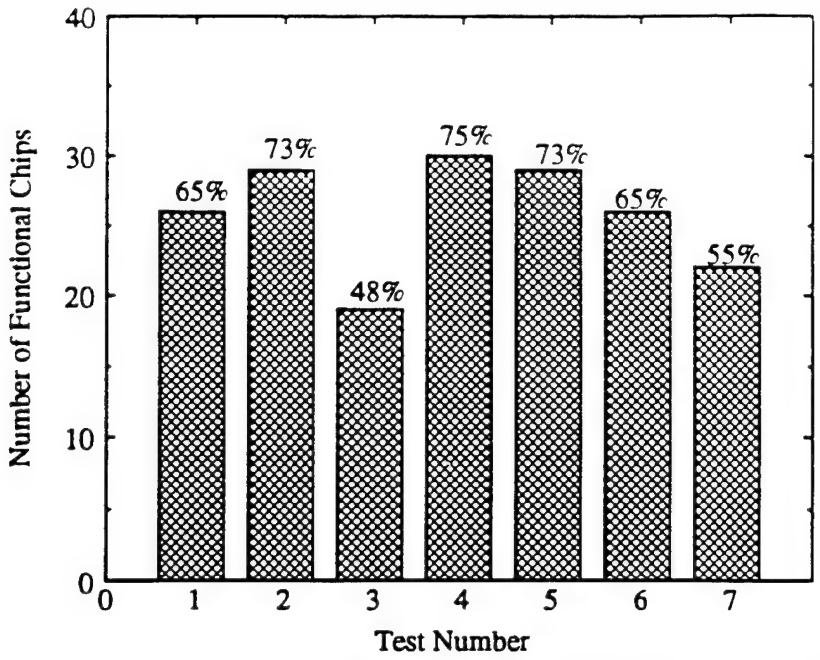


Fig. 2.25: Yield data for functional tests on the 5-port synchronous 32x64 register file.

and A are one and two diode drops above ground. Node A is pulled low by the 3-input NOR gate. When node A is lowered, node B is lowered by capacitive division as shown in Fig. (b). Adequate low-voltage noise margins can be maintained in this circuit for node B by correctly ratioing the size of diode D1 to the size of the pull-down transistor of inverter I.

2.7.1 Testing Strategy

Forty FPRF chips were fabricated, packaged and tested. Seven tests were chained together to test the functionality of different parts of the SRAM. The first four tests were designed to individually test each of the four read ports using GALPAT. The fifth, sixth, and seventh tests were each designed to simultaneously access the same address from two, three and four ports. The testing was performed using an HP 82000 IC Evaluation System. A test flow program was written to automate data collection. Each test was performed across a range of supply voltages to find the sensitivity of functionality to supply voltage. The test results are summarized in Figs. 2.25 and 2.26. Fig. 2.25 shows the number of chips that passed each of the seven functional tests. Tests number 1 through 4 exercised the individual read ports. Tests 5, 6, and 7 show the results of reading from port 2 while simultaneously accessing the same address from two, three, and four ports.

The supply voltage range over which each of the chips passed functional tests is plotted as a function of the chip number in Fig. 2.26.

2.7.2 Analysis of Failure Modes

During the initial design phase, resistive drops were considered along the cell-ground line and word line without including drops across the drivers themselves. This additional length resulted in a 60mV degradation in access transistor gate-source voltage at the end of the memory array furthest from the drivers. This caused a significant reduction in read access current, leading to non-functioning bits. The floorplan for half of the SRAM is shown in Fig. 2.27. The word-line driver numbers correspond to the test numbers shown in Fig. 2.25. The read ports that exhibited the worst yield were ports number 3 and number 1. The word drivers for these ports were also furthest away from the memory array, leading to large I-R drops along their word lines. The failing bits in these tests were located at the end of the array, furthest from the word line drivers.

In a new implementation of this register file, two of the word-line drivers were placed on the left, and two on the right, of the memory array, thereby reducing the I-R drops external to the array. Also, the new implementation used wider metal wires for these signal lines when routed across the word drivers.

Resistive drops in the power distribution network on the chip were responsible for pre-

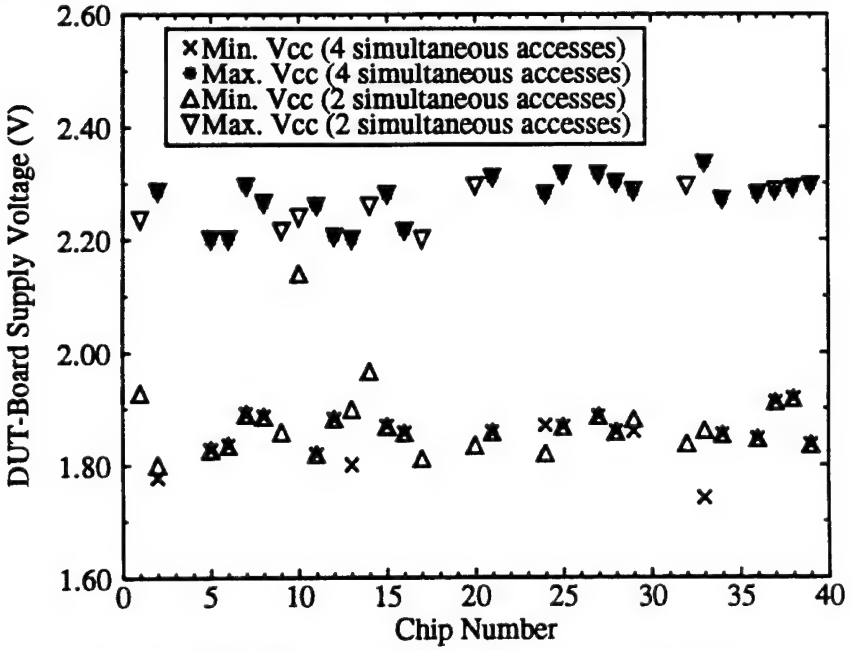


Fig. 2.26: SRAM supply voltage tolerance across chips.

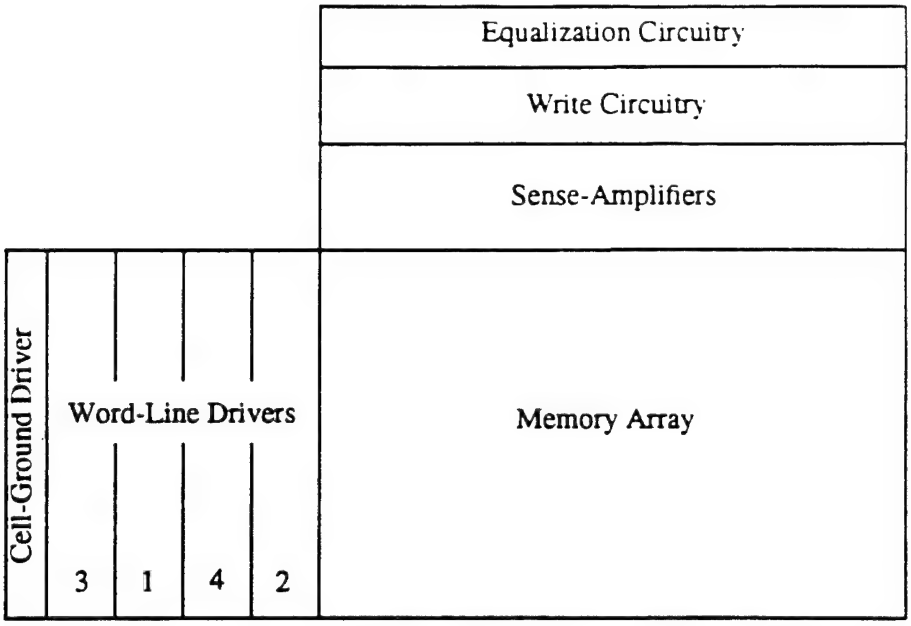


Fig. 2.27: Block diagram of the SRAM topology.

venting the memory from exhibiting functionality below 1.8V. Simulations show that the circuit block most sensitive to supply voltage is the bit-line precharge or equalization circuitry. After the circuit layout was re-examined, the power distribution to these circuits was found to be inadequate and was therefore fortified to reduce the supply voltage drops to these circuits.

The fifth, sixth, and seventh tests show that the CMMC is not immune to destructive read-out problems. When the same row is simultaneously accessed with all four addresses, the number of chips that passed GALPAT dropped from 73% for only one access to an address to 55% for four simultaneous accesses to the same address. Charge injection from the word line into the cell storage nodes was identified as the cause of this destructive read problem.

The CMMC exhibited non-destructive readout when one or two read-ports were accessed simultaneously. Each of the access transistors were 6μm wide, while the driver transistor was 2μm wide. Note that the CMMC did not show failure for a switch to driver ratio of 6:1. This compares to a 1:3 ratio that is required for the conventional memory cell. To alleviate this problem, the driver transistors in a new implementation were widened to reduce the memory cell storage node low-voltage. This increased the low-voltage noise margin, permitting more charge to be injected into the memory cell without causing destructive readout.

2.8 Conclusions

In this chapter, a new memory cell for GaAs E/D MESFET SRAMs was presented. This cell occupies a smaller cell area than a conventional 6-transistor memory cell. Faster read and write operations can be achieved for memories designed with the CMMC than for memories designed with the conventional memory cell. The biasing scheme for this cell achieves the maximum suppression of leakage currents associated with access transistors in nonselected rows. These advantages are attained while using only a single supply voltage.

Several failure mechanisms for CMMC SRAMs can be avoided by careful design. These mechanisms include leakage currents, bit-line to word-line coupling, resistive drops in the word line, and charge injection into the memory cell from the word line. The GALPAT test procedure is adequate to detect these faults as well as all stuck-at and cell coupling faults.

An experimental 1kb SRAM based on the CMMC has been designed, fabricated and tested. The demonstration vehicle proves the viability of this approach. Unfortunately, this SRAM suffered from low design yield.

A five-port synchronous SRAM was also designed, fabricated and tested. Changes were made in the design of this SRAM based on an analysis of the failure modes of the asynchronous SRAM design. Testing results for this SRAM led to a better understanding of the failure mechanisms in the CMMC.

By providing denser memories with better reliability margins and smaller access times, the CMMC should help GaAs technology to compete in the digital realm.

CHAPTER III
PROCESS TOLERANT CIRCUIT DESIGN

The asynchronous SRAM described in Chapter 2 and other early chips designed in the GaAs microprocessor project suffered from poor yield. Manufacturing defects, which cannot be controlled by the designer, were the cause of some yield problems. Other yield problems resulted from an inadequate understanding of the impact of process variations on circuit performance.

In this chapter, practical considerations needed to achieve E/D MESFET circuits with high parametric (or design) yield in the presence of large process variations are discussed. We also examine the failure modes of three different types of circuits, and present process tolerant design and analysis techniques that are well suited for each circuit type.

The manifestation of process variations in GaAs MESFETs is first described in Section 3.1. This is followed by a discussion of how these variations can cause failures in direct-coupled FET logic (DCFL) circuits. An approach is then presented for developing guidelines for the design of process-tolerant DCFL circuits.

The process intolerance of a high-drive buffer called a squeeze gate was suspected to be the source of yield loss in our early GaAs chips. In Section 3.2, the impact of global and local process variations on the noise margins of the squeeze gate and other high-drive buffers is studied. Based on this study, one buffer was found to be superior to the others, and is being used in more recent chip designs.

The major failure mode in the asynchronous SRAM, presented in Chapter 2, was stuck columns. The source of this failure was found to be process-intolerant sense-amplifier design. In Section 3.3, an algorithm for automatically designing and characterizing process-tolerant sense-amplifiers is presented. This algorithm, used in the design of the 5-port register file described in

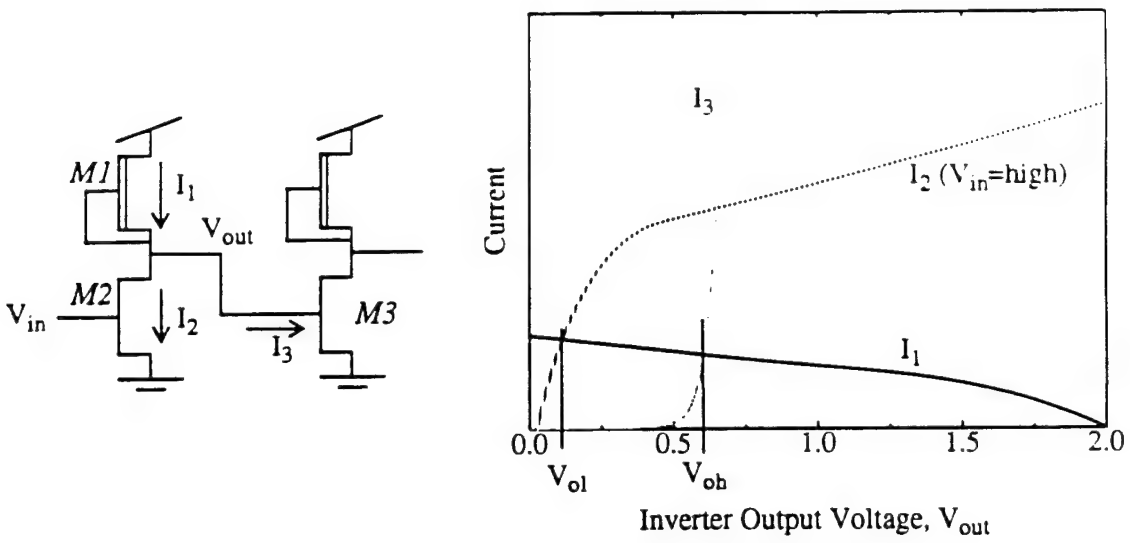


Fig. 3.1: Basic DCFL inverter and load lines.

Section 2.7, demonstrated significant yield improvements. Finally, conclusions are presented in Section 3.4.

3.1 Process Tolerance of DCFL Circuits

Variations in MESFET drain-current characteristics between devices can be modeled by variations in threshold voltage to a first-order approximation. For a uniformly doped channel, the threshold voltage of a MESFET is given by [Lon89]

$$V_T = \frac{-qN_D b^2}{2\varepsilon_s} + V_{bi} \tag{3.1}$$

where b is the thickness of the depletion layer beneath the metal-semiconductor interface, N_D is the device doping level, and V_{bi} is the built-in voltage. Factors affecting the control of threshold voltage include control of the doping profile, control of the dimensions of the device, and control of the metal-semiconductor work-function.

3.1.1 Preliminary Definitions

The basic DCFL inverter and the associated load lines are shown in Fig. 3.1. The output-high voltage, V_{out}, of inverter M1-M2, is determined by the load current, I_1, and the gate-source

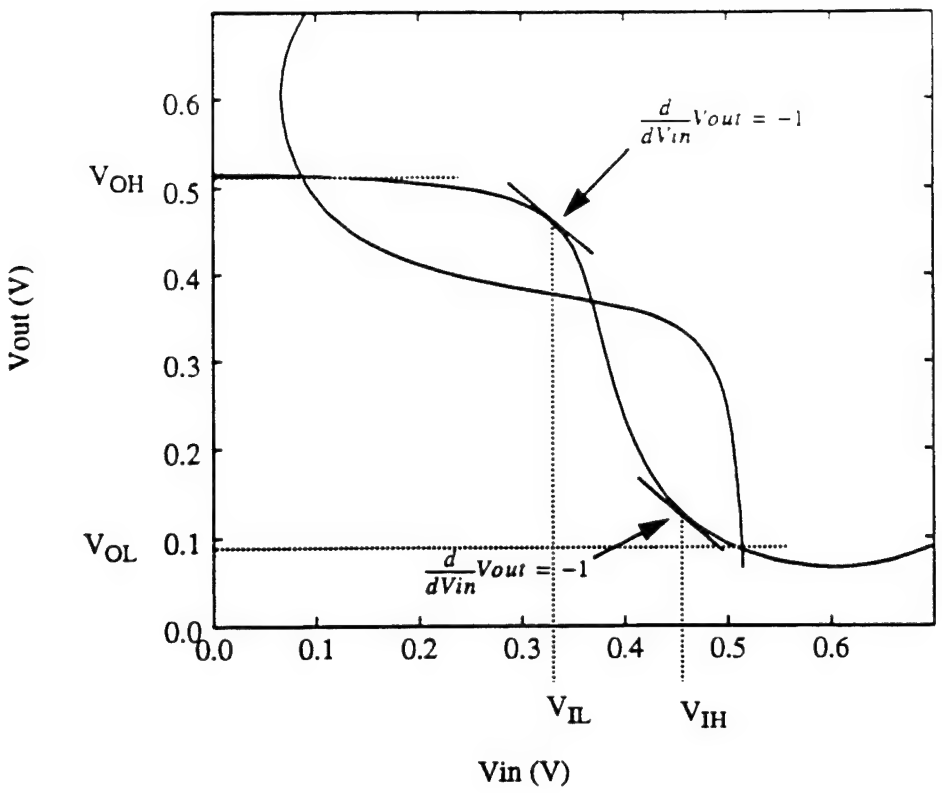


Fig. 3.2: DCFL inverter transfer characteristic, and the noise margin definitions.

Schottky diode of transistor M3. The output-low voltage of the inverter is determined by the ratio, β, of the dimensions of the pull-down transistor, M2, to the pull-up transistor, M1.

$$\beta = \frac{(W/L)_{M2}}{(W/L)_{M1}} \tag{3.2}$$

Although the ratio is a dimensionless quantity, β ratios make sense only within the context of a fixed-L device, since short-channel effects tend to dominate the behavior of sub-micron devices.

In this discussion, we will use the following definitions of noise margins.

$$NM_H = V_{OH} - V_{IH}, \quad \text{and} \tag{3.3}$$

$$NM_L = V_{IL} - V_{OL} \tag{3.4}$$

where V_{IL} and V_{IH} are the input-low and input-high voltages, given by the $\frac{d}{dVin}Vout = -1$ definitions shown in Fig. 3.2; V_{OL} and V_{OH} are the output-low and output-high voltages as defined

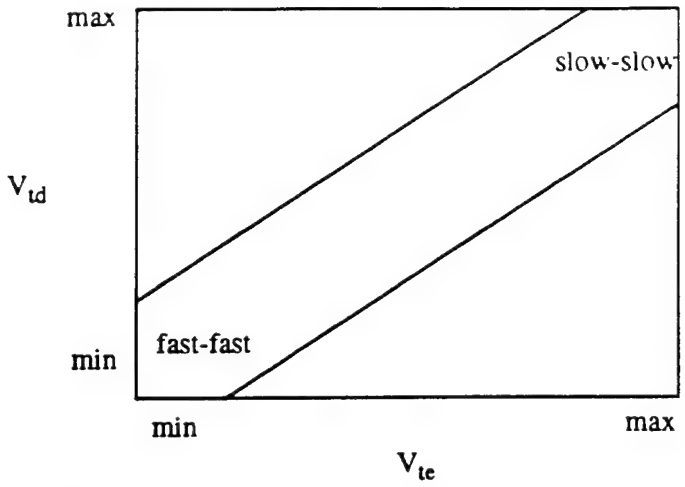


Fig. 3.3: V_{te}-V_{td} plot of inverter pull-up and pull-down thresholds.

graphically in the figure. As long as these noise margins are positive, the gain of the next logic stage ensures that the signal levels are restored.

The threshold voltages of enhancement and depletion transistors are normally distributed, having probability density functions given by

$$f_e(V_{te}) = \frac{1}{\sigma(V_{te})\sqrt{2\pi}} \cdot exp\left[-\frac{(V_{te}-V_{teo})}{2\sigma(V_{te})^2}\right], -\infty \le V_{te} \le \infty \tag{3.5}$$

$$f_d(V_{td}) = \frac{1}{\sigma(V_{td})\sqrt{2\pi}} \cdot exp\left[-\frac{(V_{td}-V_{tdo})}{2\sigma(V_{td})^2}\right], -\infty \le V_{td} \le \infty \tag{3.6}$$

where V_{teo} and V_{tdo} are the nominal centers of the V_{te}-V_{td} process space, and $\sigma(V_{te})$ and $\sigma(V_{td})$ are standard deviations of the enhancement and depletion transistor threshold voltages. Many GaAs manufacturers [Wil94] report that these variations tend to track one another locally, so an inverter with a high value of V_{te} will also have a high (less negative) V_{td}, and inverters with a low value of V_{te} will also have a low (more negative) value of V_{td}. Fig. 3.3 shows that inverter pull-up and pull-down transistor thresholds will define points that lie in a band centered about a line extending from the slow-slow to fast-fast process corners. These corners are referred to as slow-slow and fast-fast because larger threshold voltages lead to smaller transistor currents, while smaller threshold voltages lead to larger transistor currents. Larger currents, of course, lead to

smaller signal delays.

3.1.2 Simplifying Assumptions

A few simplifying assumptions are made before analyzing the impact of threshold voltage control on yield.

1. All processing variations can be modeled by variations in threshold voltage.

2. The circuits are comprised of only DCFL circuits. (The process tolerance of feedback-FET-logic buffers, commonly used in VLSI circuits, is discussed in the next section.)

3. Temperature across the die is uniform. Variations in temperature across a die are beyond the scope of this analysis.

4. The enhancement and depletion transistor threshold voltages track each other locally (as in Fig. 3.3).

3.1.3 Parametric Yield Formulation

Using these assumptions, the circuit shown in Fig. 3.4 can be used to analyze a circuit's parametric yield. In this circuit, we have a driver-inverter with thresholds V_{te1} and V_{td1}, and a driven inverter with thresholds V_{te2} and V_{td2}. Since local enhancement and depletion thresholds track each other, any discussion of variations in V_{te1} and V_{te2} will imply variations in V_{td1} and V_{td2}. The noise margins of a functional circuit need to be greater than a minimum safety margin. This safety margin provides allowance for noise from ground bounce, capacitive coupling from signal lines, and noise from off-chip sources.

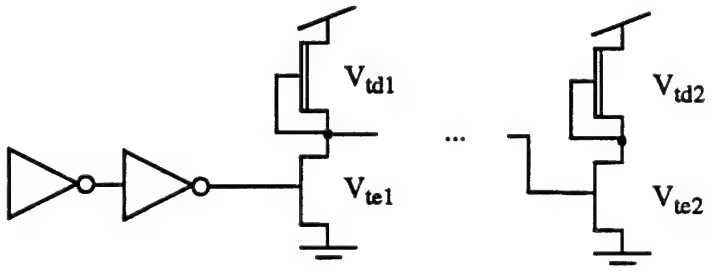


Fig. 3.4: Model for finding non-functional circuits.

Formally, the functionality of a DCFL circuit consisting of a driver inverter with enhancement threshold V_{te1}, and a driven inverter with enhancement threshold V_{te2}, can be defined as

$$fnl(V_{te1}, V_{te2}) = 1$$

$$\text{iff} \qquad NM_L \geq NM_{MIN}$$

$$\text{and} \qquad NM_H \geq NM_{MIN} \quad ;$$

$$fnl(V_{te1}, V_{te2}) = 0 \text{ otherwise.}$$

For technologies such as ECL, a minimum noise margin of 50mV is considered adequate[Zde94]. Voltage drops in the distribution of ground cause additional losses in low-voltage noise margins. Thus, a minimum safety margin of 75mV will be assumed adequate for DCFL circuits.

The parametric yield of a circuit containing a string of two DCFL logic gates is given by

$$Y = \int_{V_{te1}} f_e(V_{te1}) \int_{V_{te2}} f_e(V_{te2}) fnl(V_{te1}, V_{te2}) \, dV_{te1} dV_{te2} \tag{3.7}$$

where f_e is the probability density function defined in (3.5). Since the function $fnl(V_{te1}, V_{te2})$ does not have a closed form expression, the integral can more easily be computed in its discrete form as

$$Y = \sum_{V_{te1} = minV_{te}}^{maxV_{te}} Pr\left(V_{te1} - \frac{\Delta}{2} \leq V_{te} \leq V_{te1} + \frac{\Delta}{2}\right)$$

$$\cdot \sum_{V_{te2} = minV_{te}}^{maxV_{te}} Pr\left(V_{te2} - \frac{\Delta}{2} \leq V_{te} \leq V_{te2} + \frac{\Delta}{2}\right) \cdot fnl(V_{te1}, V_{te2}) \tag{3.8}$$

where $fnl(V_{te1}, V_{te2})$ is evaluated on a grid of resolution Δ over the (V_{te}, V_{td}) space, and

$$Pr(\alpha \leq V_{te} \leq \gamma) = \int_{\alpha}^{\gamma} f_e(V_{te}) \, dV_{te} \tag{3.9}$$

3.1.4 Failure Modes for DCFL Circuits

Before applying these yield equations, it is worthwhile to examine the failure modes of

DCFL circuits. Contour graphs for V_{IH}, V_{OH}, V_{IL}, and V_{OL} as functions of V_{te} and V_{td} are shown in Fig. 3.5. The data for these graphs were generated for an inverter having β=8, at the lower end of the commercial operating temperature range, -40°C. At this temperature and β value, the output-high voltage is no less than 700mV, while the required input-high voltage is not much greater than 550mV. Thus, the minimum 50mV high-voltage noise margin is easily met.

However, the noise margins are smaller for the output-low voltage. At the same temperature and β value, the output-low voltage can be as high as 125mV. The input-low voltage requirement, shown in Fig. 3.5(d), can be as low as 150mV. Thus, the minimum noise margin across process variations between a driver and a driven gate is only 25mV. This margin is too low, and can lead to circuit failure.

The low-voltage noise margin, as illustrated above, is problematic for DCFL circuits. This is not the case for the high-voltage noise margin. At increased temperatures, the low-voltage noise margin is reduced further for multi-input NOR gates. Fig. 3.6(a) shows a four-input NOR gate, with three inputs driven by a logic low. Transfer characteristics for this circuit at -40°C, 80°C, and 120°C are shown in Fig. 3.6(b). The leakage currents flowing from the output-node to ground through the three inputs that are driven low reduce the input-high voltage required at the fourth input to trip the inverter. As leakage currents increase with temperature, the trip-point of the inverter shifts toward ground, further reducing the input-low voltage and thus the low-voltage noise margin. Fig. 3.6(c) summarizes the effect of a NOR gate's fan-in on its transfer curve. As the number of inputs is increased from 1 to 8, a dramatic shift in the transfer curve, and hence in the noise margin takes place. These curves were generated using a β ratio of 12 at 80°C. Because of this effect, the fan-in of DCFL gates is typically restricted to 4. As the β value of the driver inverter is increased, the output-low voltage is decreased, which reduces the problematic leakage currents. The effect of β on the NOR gate transfer curve is shown in Fig. 3.6(d).

3.1.5 Design β Choice

In the last section, DCFL circuits were shown to exhibit failures resulting from inadequate output-low voltage noise margins. High fan-in, high temperature, and low β values cause DCFL NOR gate transfer characteristics to shift toward ground, resulting in reductions in the low-voltage

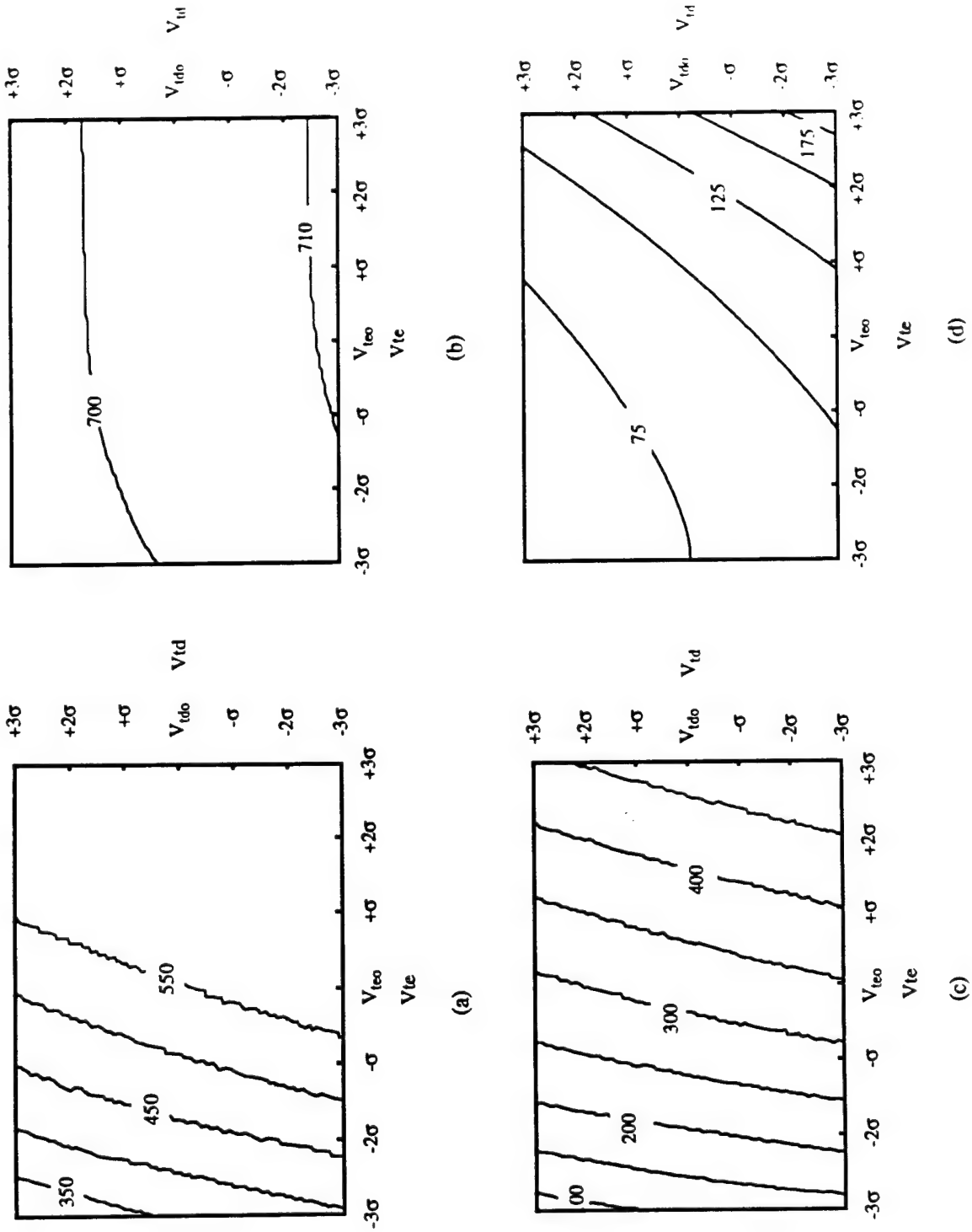


Fig. 3.5: Impact of process variations on the parameters that dictate noise margins.
(a) V_{IH} (mV) variations with process ; (b) V_{OH} (mV) variations with process;
(c) V_{IL} (mV) variations with process; (d) V_{OL} (mV) variations with process.

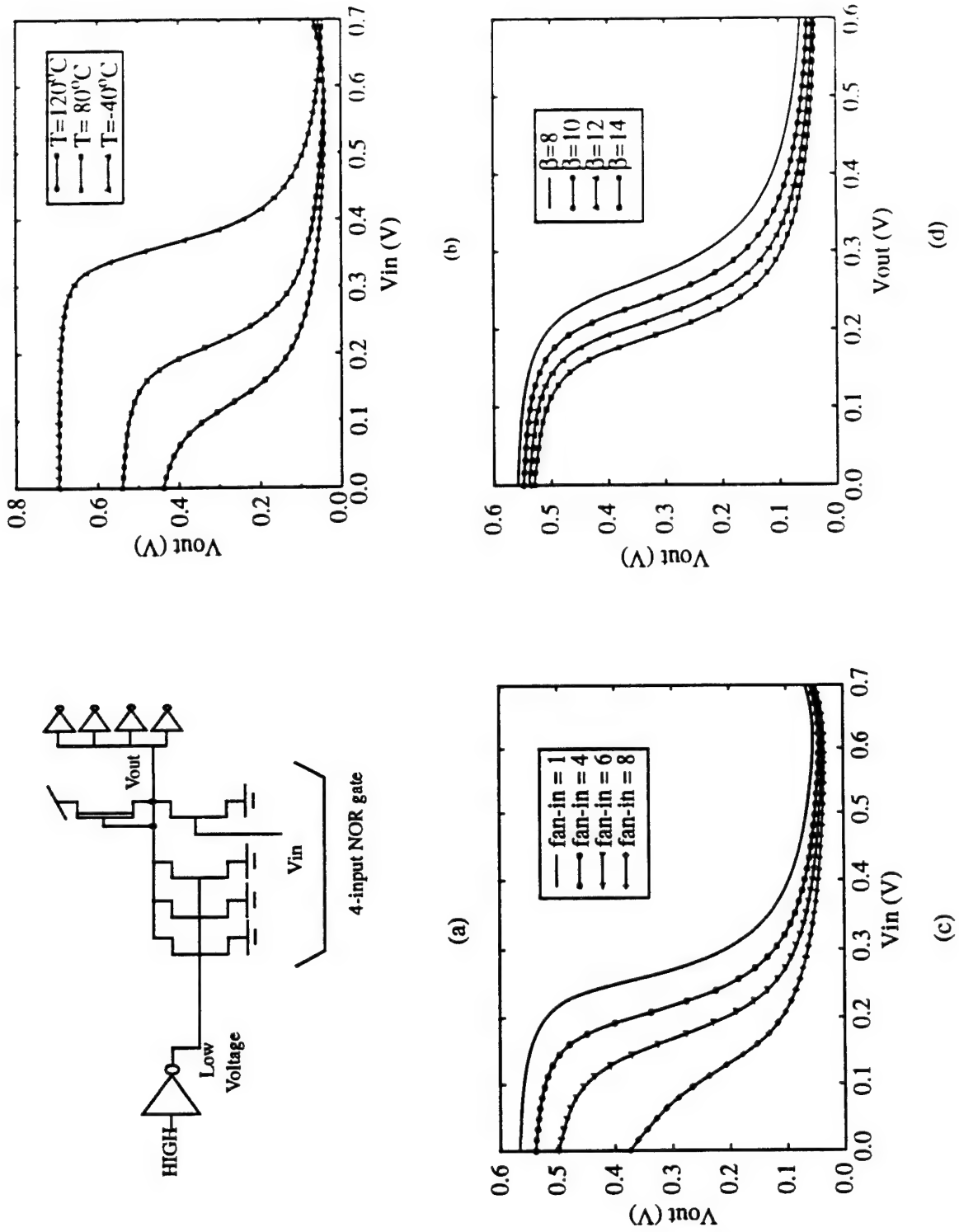


Fig. 3.6: Impact of temperature, fan-in, and b on NOR gates.
(a) A four-input NOR gate with 3 inputs driven LOW; (b) transfer characteristics of this NOR gate at various temperatures; (c) NOR gate transfer curves for different fan-ins at T=80°C, β=12; (d) 4-input NOR gate transfer curve for different β values at T=80°C, with all inputs tied low except for one.

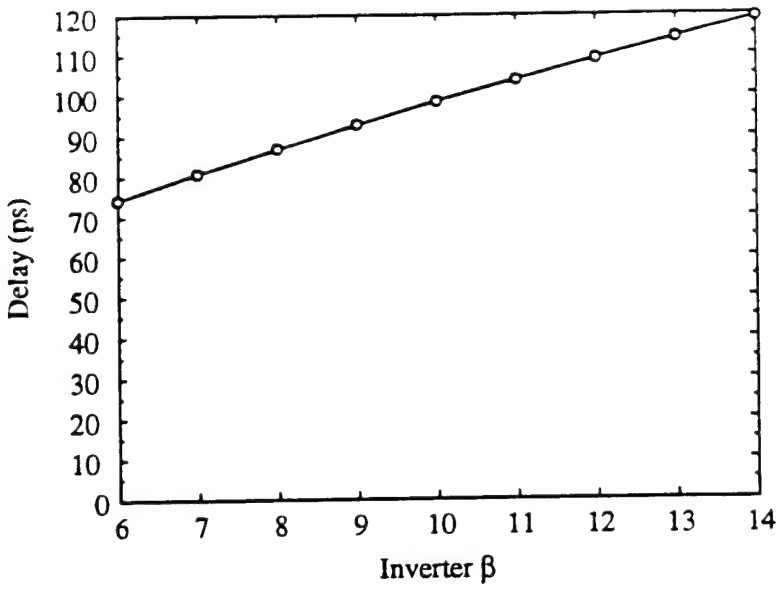


Fig. 3.7: The per-stage delay of an unloaded chain of DCFL inverters as a function of inverter β ratio, for a fixed pullup device size.

noise margins. Of these variables, the circuit designer has control of only the β value.

The effect of inverter β on propagation delay is shown in Fig. 3.7. This figure shows the intrinsic delay of a DCFL inverter as a function of β. The difference between the delays of different β-inverter chains in real circuits that are loaded with routing parasitic capacitances would be smaller than that shown in the figure.

To aid in the calculation of parametric yield, two scripts were written. The first script exercised an HSPICE simulation to gather and extract V_{OH}, V_{IH}, V_{OL}, and V_{IL} information for a specified β value and temperature. Since the low-voltage noise margin is of concern, the circuit used to gather the V_{IH} and V_{IL} data was a DCFL NOR gate with a fan-in of 4, at 100°C. These data were gathered across V_{te} and V_{td} increments of 5mV. A second script was written to use the data generated by the first script to calculate the yield of a circuit using (3.8) and (3.9).

Results of this study are shown in Fig. 3.8, where the yield of a circuit consisting of 10K DCFL gates is plotted as a function of threshold voltage control for different design β values. Even though a lower values of β result in a smaller signal swings and a somewhat larger output-low voltages, circuits with higher βs are actually somewhat less tolerant of process variations due to their lower input-low voltages. Thus, the choice of βs with a higher tolerance to low-voltage noise margins also leads to faster circuits.

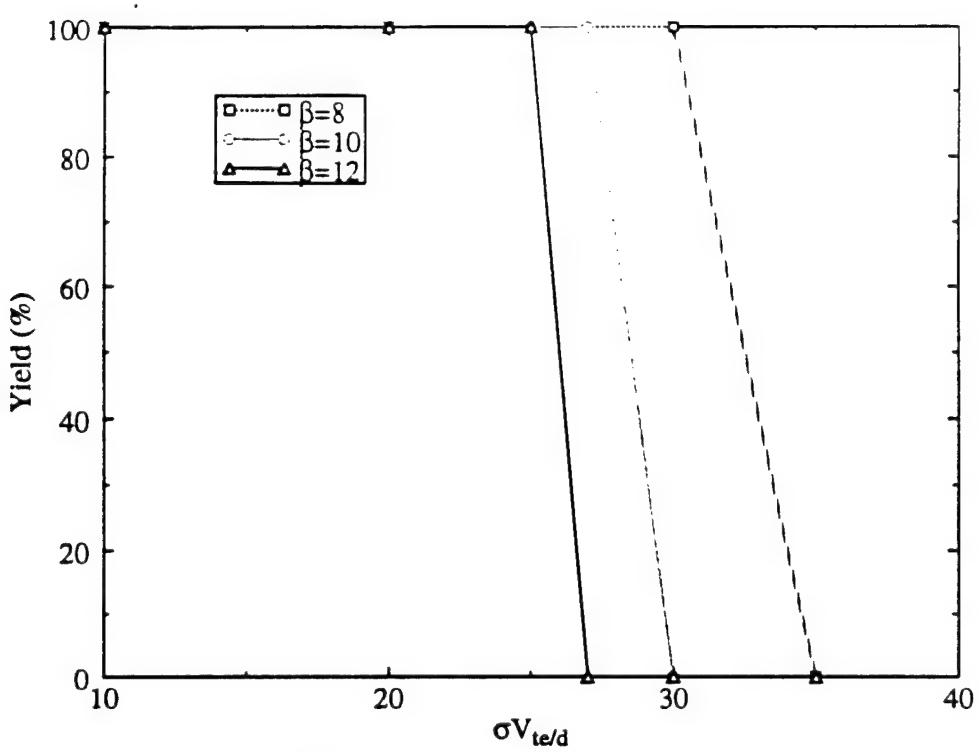


Fig. 3.8: Circuit yield for a 10K-gate DCFL circuit as a function of transistor threshold voltage standard deviation for different β values.

This graph shows a sharp drop in yield above a $\sigma V_{te/d}$ of 27mV. Above this variation, enough circuits with small V_{IH} values enter the process space to result in a high probability of failures. Since the parametric yield stays flat at 100% until a critical threshold variation, this graph also describes the parametric yield for much larger levels of integration.

3.1.6 Applications

The probabilistic model of DCFL circuit yield has been used to study the impact of design β selection on circuit yield. The software tool and design methodology developed for this analysis can be extended to answer several process-related questions, such as:

1. What are the optimal target enhancement- and depletion-transistor thresholds for a given an operating environment and threshold voltage spread?

2. What process control is needed to achieve a certain minimum yield for circuits with a given integration level?

As short channel effects dominate the characteristics of devices with ever-decreasing

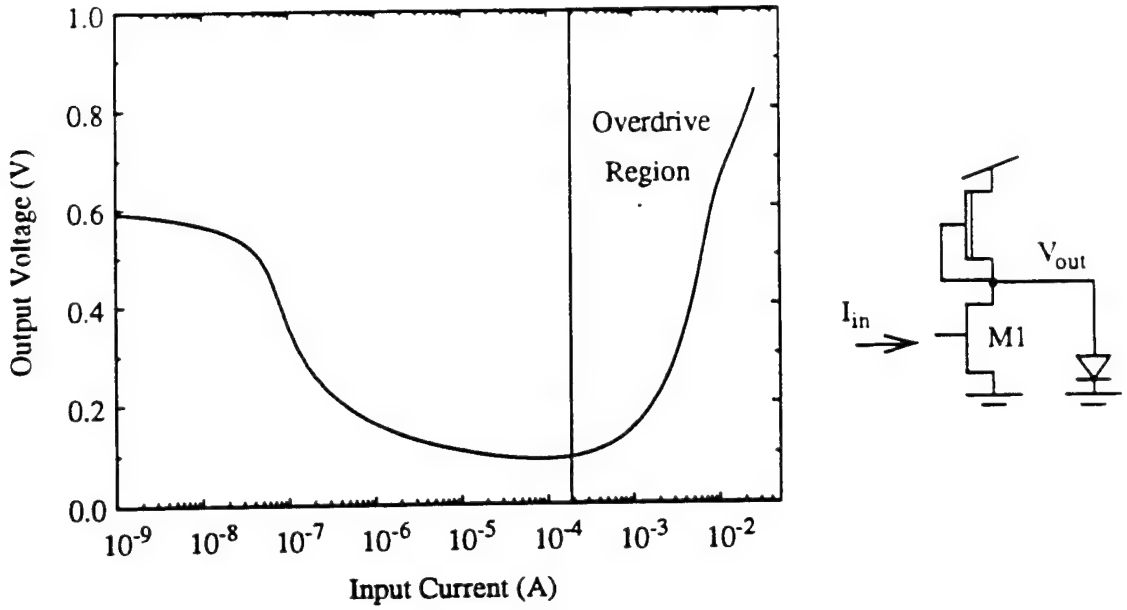


Fig. 3.9: DCFL inverter transfer characteristic.

geometries, such an analysis will become a necessary component of any circuit-design strategy.

3.2 Process Tolerance of Super-Buffers Using Feedback.

The MESFET's Schottky-diode gates can cause an overdriving condition when a buffer drives a large current onto a line to achieve a short delay. Although a large current is needed to charge the line quickly, it may be too large for the static current sinking capability of the gates on the line. Fig. 3.9 shows the inverter transfer characteristic for a minimum-sized gate. Above input currents of 200μA, the input gate current to MESFET M1 flows not only through the source, but also out the drain, thereby increasing the output voltage to a logic ONE, when a logic ZERO was desired.

Large buffer drive currents are necessary to achieve acceptable edge rates on signal lines. For example, 5mA of driver current is needed to achieve a 200ps rise time for a 0.5V signal swing on a 2pF line. This amount of current forces the driven DCFL inverter well into the overdrive region. In addition to causing a logic problem, this overdrive condition also wastes a considerable amount of static power.

Many variations of super buffers using feedback have been proposed in the literature

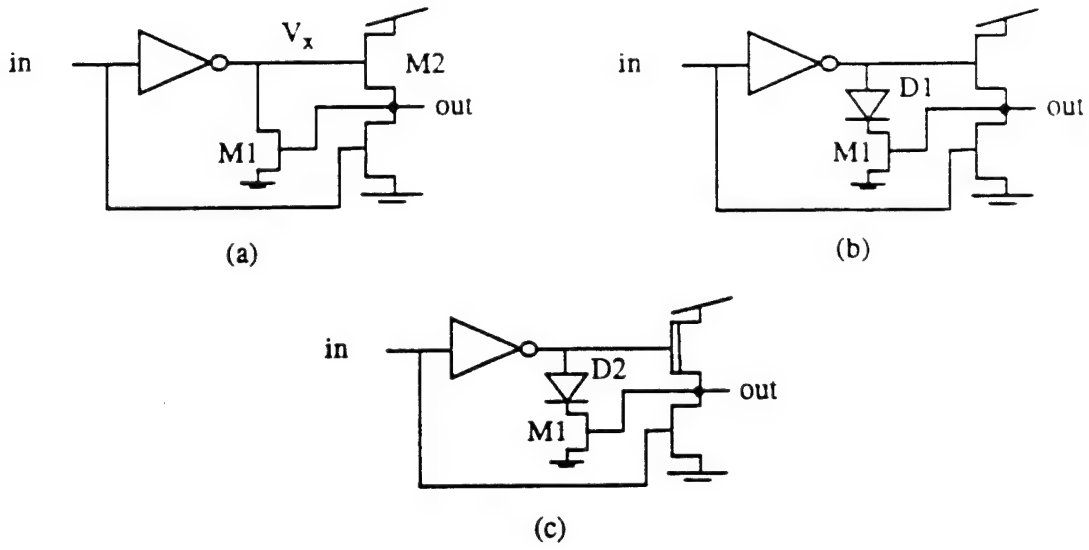


Fig. 3.10: Alternate super-buffer implementations using feedback.
(a) squeeze gate; (b) FFL gate; (c) squirt gate.

[Ful91], [Lar90] to solve this problem. These implementations, shown in Fig. 3.10, are referred to as (a) a squeeze gate, (b) a feedback-FET-logic (FFL) gate, and (c) a squirt gate.

Fig. 3.11 illustrates the operation of the squeeze gate. When the input of the buffer is brought low, the intermediate node voltage, V_x, starts to rise. This turns on the output pullup transistor, M2, which in turn causes a surge of current to charge the output. As the output voltage starts to rise, it turns on the feedback device, M1. This lowers the intermediate node voltage, V_x, limiting the static current drawn by the buffer. The feedback action also prevents the overdrive problem described above, thus providing a power-efficient approach to quickly charging a highly capacitive line.

The squeeze buffer is the simplest of these buffers, using only one feedback transistor. The FFL and squirt buffers also use a diode in the feedback path to increase the signal swing of the intermediate node. These buffers also differ in the type of pullup devices used; the FFL and squeeze buffers use enhancement output pullup devices, while the squirt buffer uses a depletion pullup. These buffers all exhibit a large transient demand for supply current when switching, unlike DCFL which is a current-steering logic.

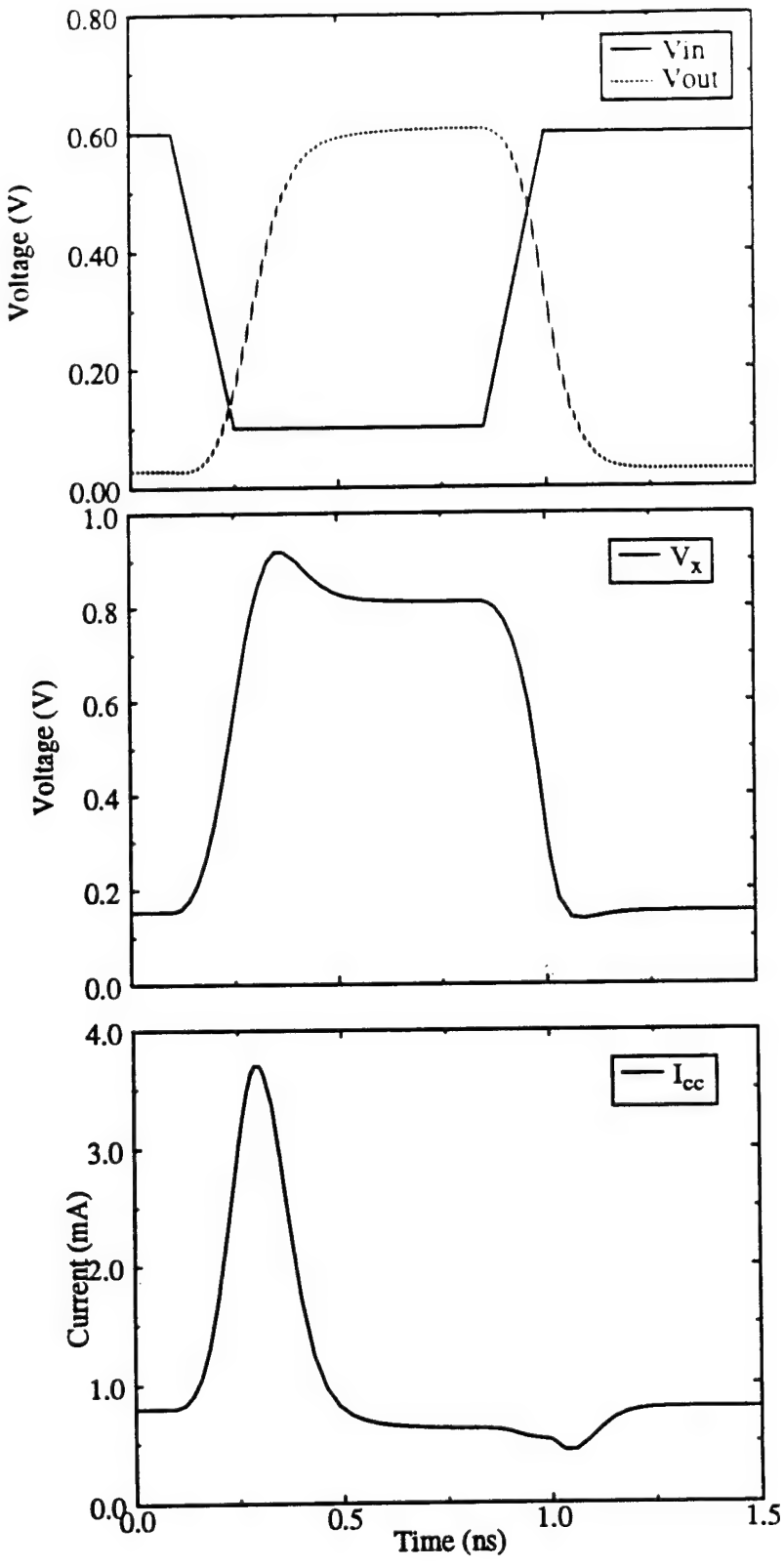


Fig. 3.11: Timing diagram of the squeeze buffer operation.
(a) input and output waveforms; (b) intermediate node voltage; (c) supply current.

3.2.1 Process Tolerance

The tolerance of these buffers to local and global process variations is an important consideration in their comparison. The approach used to compare the process tolerance of these feedback buffers is similar to the approach used in the previous section for analyzing DCFL circuits.

The output-low voltages of all of these buffers are a few tens of millivolts, and hence, they achieve adequate low-voltage noise margins. The high-voltage noise margins, therefore, differentiate the process tolerance of the buffers. Fig. 3.12(a) shows, as a function of process, the input-high voltage requirements for a DCFL gate with unity fan-in and a β of 8, at a temperature of 120°C. Figs. 3.12(b), (c), and (d) show the output-high voltages of the various buffer types.

Over this process space, the DCFL inverter requires an input-high voltage as high as 560mV. Assuming that the enhancement and depletion thresholds track each other locally, the squeeze buffer may be capable of providing only 500mV for the output-high voltage, resulting in a negative noise margin. Signal coupling between lines, resistive drops along the supply voltage rail, resistive drops along the signal lines, and local variations within the squeeze buffer further degrade the high noise margin.

The squirt and FFL buffers display more tolerance to global process variations than does the squeeze buffer. Both buffers exhibit only a 30mV change in output-high voltage across the process space. The only difference between the FFL buffer and the squeeze gate is the presence of a diode in the feedback path. The addition of this diode ensures that the gate terminal of the super-buffer pullup transistor remains at least one diode drop above ground.

Fig. 3.13 shows the effect of local process variations in the buffers on the output-high voltage. The worst-case scenario occurs when there is a weak pullup device and a strong feedback device. The output-high voltage is plotted as a function the relative strength of the feedback transistor to the pullup device, measured in mV of threshold voltage. As expected, the squeeze buffer displays a much greater sensitivity to local threshold voltage mismatch, while the squirt and FFL gates show little dependence on local mismatch. The squeeze gate exhibits 0.5mV output-high voltage degradation for every 1mV of local mismatch. Thus, a 30mV mismatch in local threshold voltage within the squeeze buffer results in an additional 15mV loss in output-high voltage margin. The squeeze gate failures become more pronounced at shorter channel lengths where the feedback

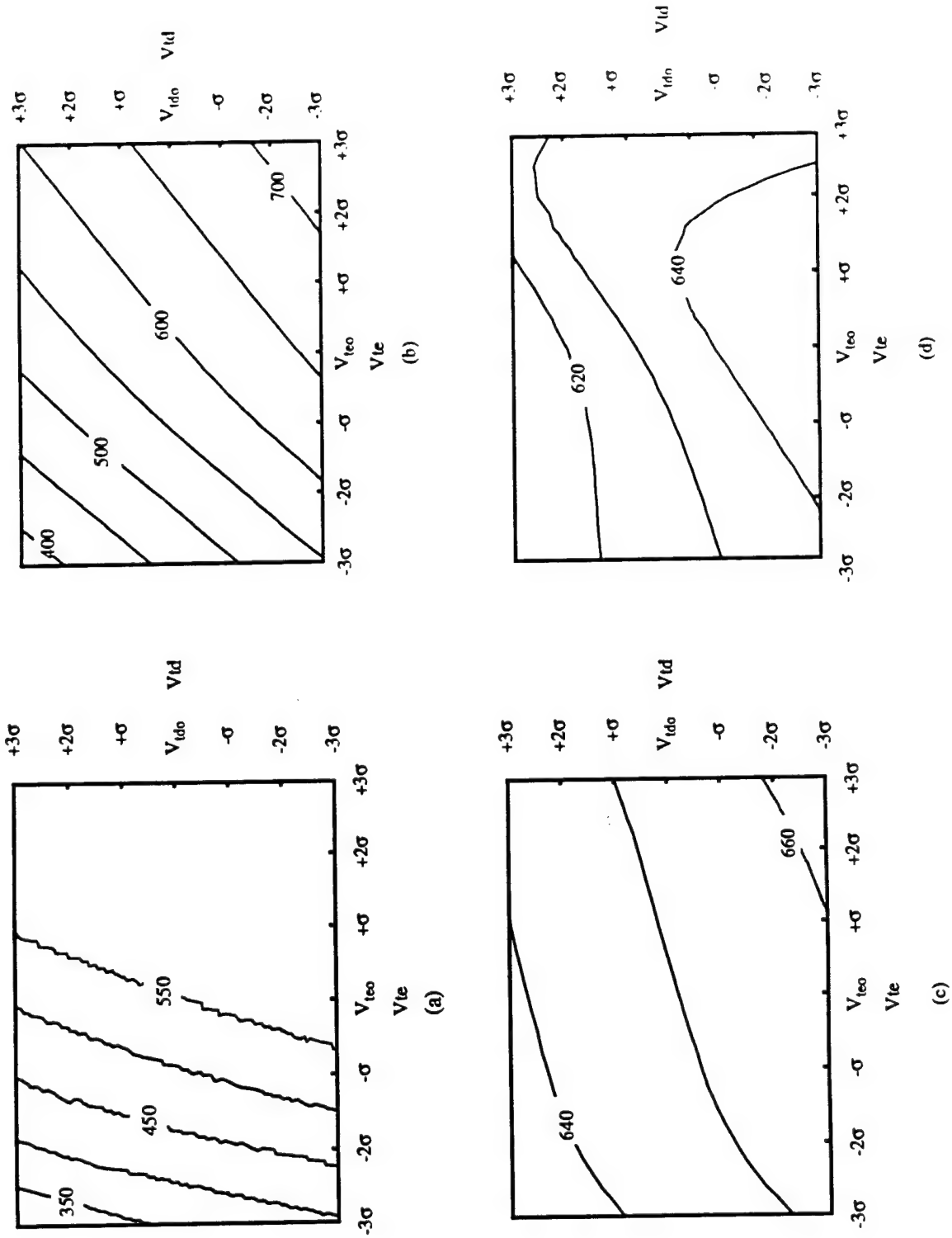


Fig. 3.12: Comparison of the process tolerance of the squeeze, squirt, and FFL buffers. (a) DCFL input-high voltage requirement; (b) squeeze buffer output-high voltage; (c) squirt buffer output-high voltage; (d) FFL buffer output-high voltage.

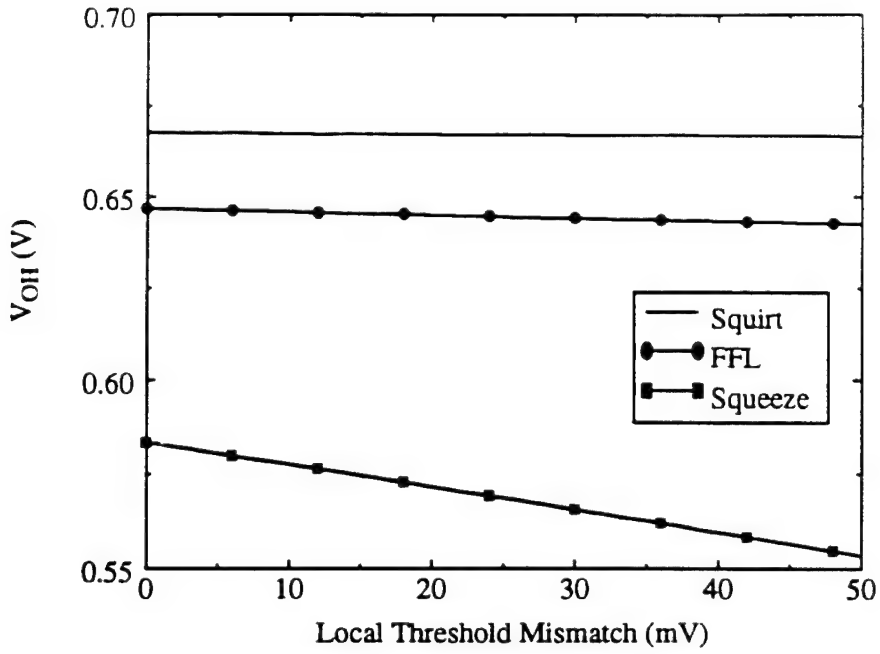


Fig. 3.13: Impact of local threshold voltage mismatch on buffer output-high voltage.

transistor supports a much smaller drain-source voltage drop.

3.2.2 Power Comparison

The power dissipation of all three buffers is a function of the diode load, the size of the feedback transistor, the size of the capacitive load, and the frequency of switching. Figs. 3.14 and 3.15 show contour plots of the power dissipations and output-high voltages as functions of the feedback transistor size, and output load. The buffers used in these simulations were driving similar capacitive loads at a 250MHz toggling frequency. The FFL buffer and squeeze buffer that are compared in this simulation use the same transistor sizes, while the squirt buffer transistor sizes were chosen to achieve similar delays to the FFL and squeeze gate over the range of simulated loads.

The power dissipations of the different buffers must be compared for equal high-voltage noise margins. For example, when driving a 1,000μm-gate load to achieve a 0.62V output-high voltage, a squeeze gate requires a 5μm wide feedback transistor, while a FFL gate requires a 7.8μm wide feedback transistor. The powers dissipated by the gates are roughly equal. Thus, the FFL gate achieves enhanced tolerance to process variation with no additional power requirement.

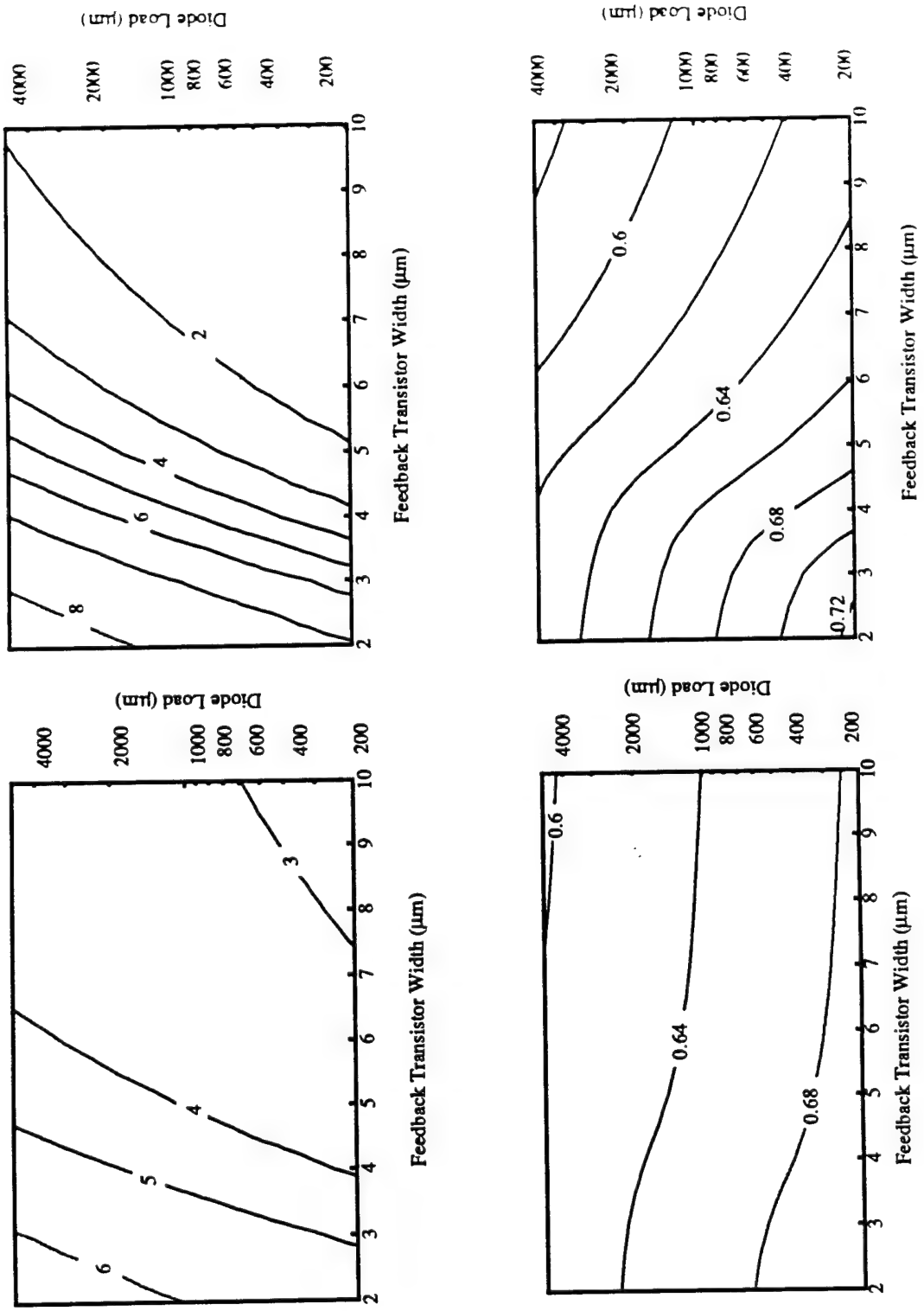


Fig. 3.14: Power dissipation and output-high voltage contour plots for the FFL, squeeze and squirt buffer as a function of feedback and load transistor sizes. (a) squirt buffer power dissipation (mW); (b) FFL buffer power dissipation (mW); (c) squirt buffer output-high voltage (V); (d) FFL buffer output-high voltage (V).

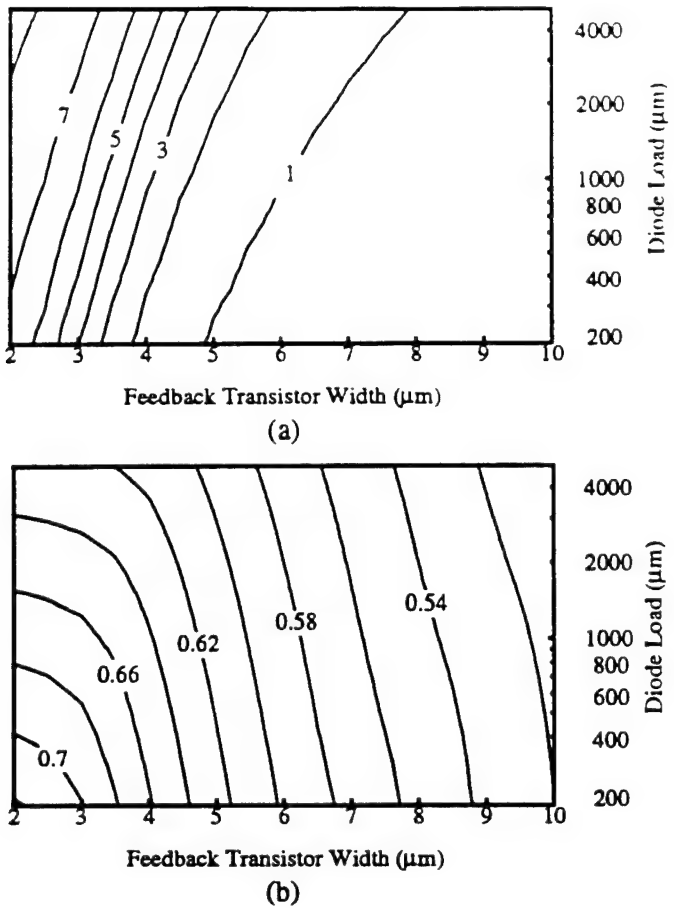


Fig. 3.15: Power dissipation and output-high voltage contour plots for the squeeze gate. (a) power-dissipation (mW); (b) output-high voltage (V).

Since the depletion load transistor on the squirt gate can not be turned off, it dissipates much more power than the FFL gate when achieving the same output-high voltages. Thus, based on local and global process variation tolerance, and on power dissipation considerations, the FFL gate is superior to the squirt and squeeze buffer gates.

3.2.3 Automatic Buffer Sizing

Automatic buffer sizing procedures used in CMOS reduce the delays of signals along critical paths. An automatic buffer sizing algorithm for E/D MESFET feedback-type buffers must also incorporate noise margin considerations. The power dissipation and output-high voltage graphs (Figs. 3.14 and 3.15) show that there is considerable room for trade-offs between noise margin and power dissipation. This trade-off is shown more clearly in Fig. 3.16, where the data

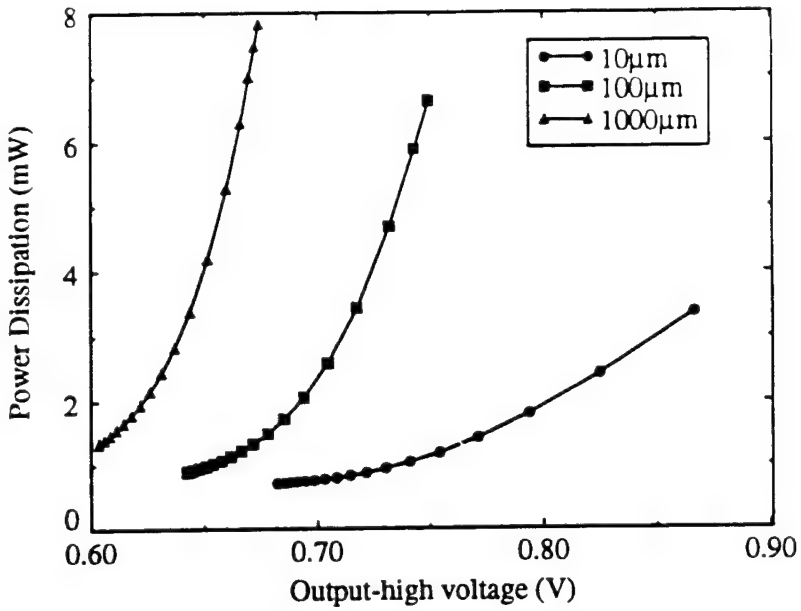


Fig. 3.16: Power-noise margin trade-off for an FFL buffer driving different sizes of diode loads.

plotted in Fig. 3.14 (b) and (d) are combined, showing the power dissipation required to achieve a given output-high voltage for various diode loads. Each of these curves is generated by varying the feedback transistor size.

In a noise-margin driven automatic buffer sizing algorithm, a minimum output-high voltage noise margin, NM_H, would be specified first. The output-high voltage, V_{OH}, for a given buffer would then be calculated based on the input-high voltage V_{IH} requirement of the driven DCFL gates. Next, the feedback transistor would be sized to achieve the specified noise margin. The buffer delay is virtually independent of the feedback transistor size. Therefore, a delay-driven approach can be used to size the buffer, followed by a noise-margin approach to size the feedback transistor.

3.3 Process Tolerant Sense-Amplifier Design

The largest source of yield problems for the asynchronous SRAM design presented in Chapter 2 was stuck columns. These columns were pinned at a zero during readout. The source of this problem was determined to be a process-sensitive sense-amplifier. The sense-amplifier is shown here in Fig. 3.17. This circuit consists of five stages. The first stage translates and ampli-

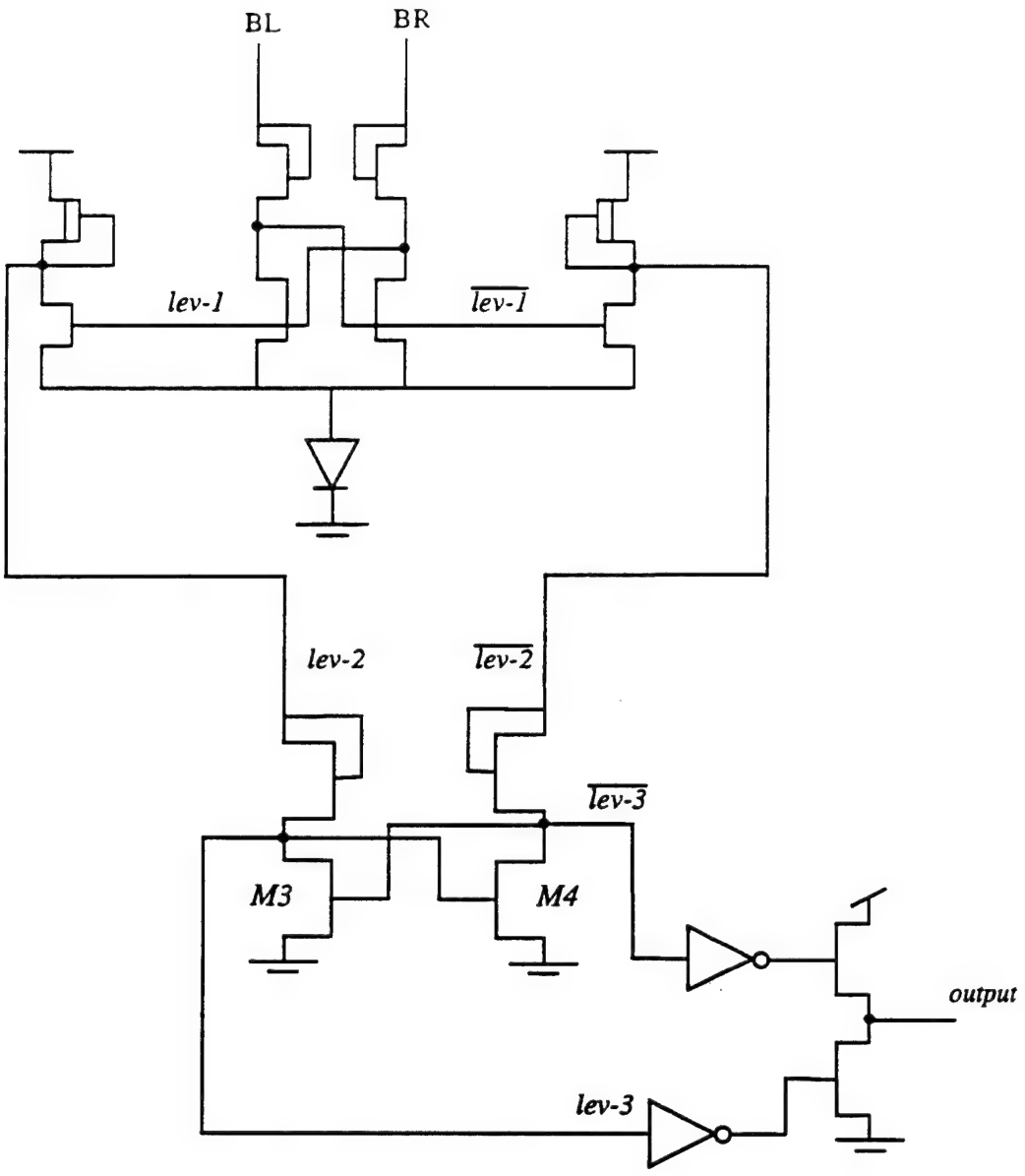


Fig. 3.17: Original (process-sensitive) sense-amplifier design.

fies the complementary bit-line signal swing. The outputs of this stage, *lev-1* and $\overline{lev\text{-}1}$, are inputs to a pair of DCFL inverters that have their pull-downs tied to one diode drop above ground. The outputs of these inverters, *lev-2* and $\overline{lev\text{-}2}$, are then level shifted to produce DCFL levels, *lev-3* and $\overline{lev\text{-}3}$.

The sense-amplifier design is a sequence of level shifters and signal amplifiers. Each stage places a different requirement on the signal levels of the previous stage. For this circuit to

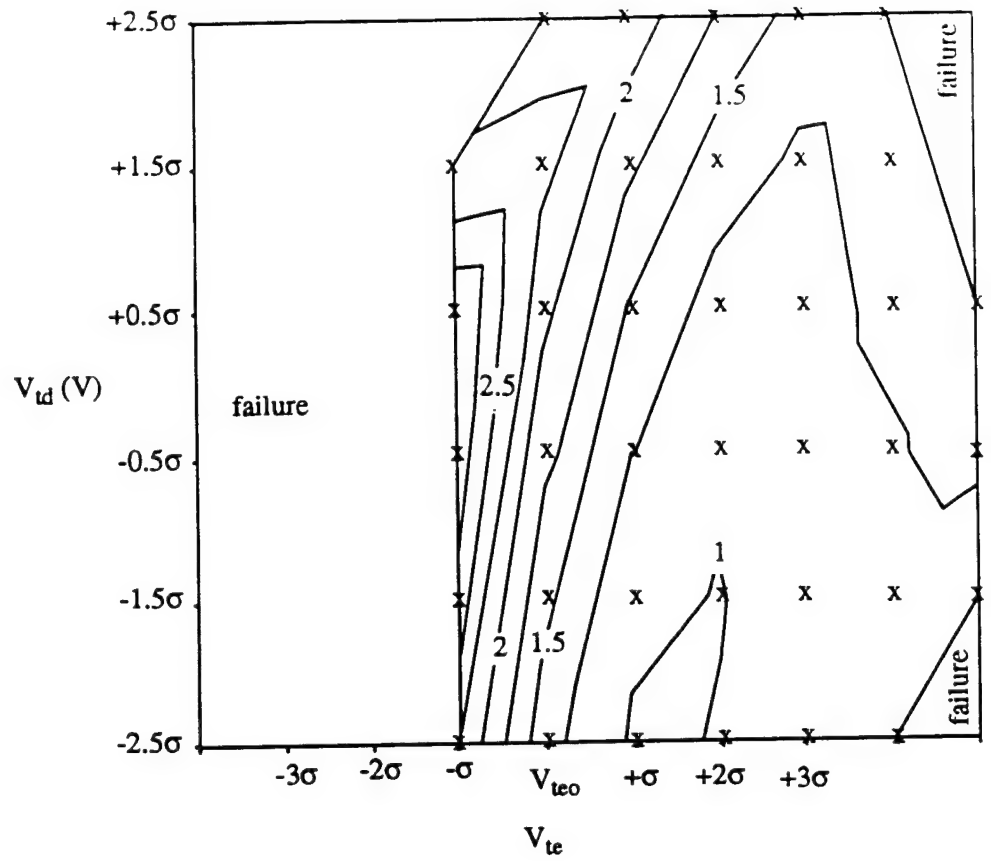


Fig. 3.18: Simulated access time of the asynchronous SRAM as a function of enhancement and depletion threshold voltages.

work properly, each stage must be tolerant of variations in the common-mode voltages of the previous stage. An inadequate understanding of the sensitivity of the various stages to process variations led to the yield problems associated with the sense-amplifier in the asynchronous SRAM.

Fig. 3.18 is a contour graph showing the memory delay, in nanoseconds, as a function of the enhancement and depletion threshold voltages for the original sense-amplifier design. The points in the plot marked with an "X" are the discrete simulation points that produced functional sense-amplifiers. Below $V_{te}=V_{teo}-\sigma$, simulations show that *lev-3* and $\overline{lev\text{-}3}$ were too high to allow a ONE to be passed to the output. The manner in which the sense-amplifiers failed in simulations is consistent with the observed behavior during testing, which was that all failed columns were stuck-at ZERO.

Another problem pointed out by the access time contour graph is a substantial variation in

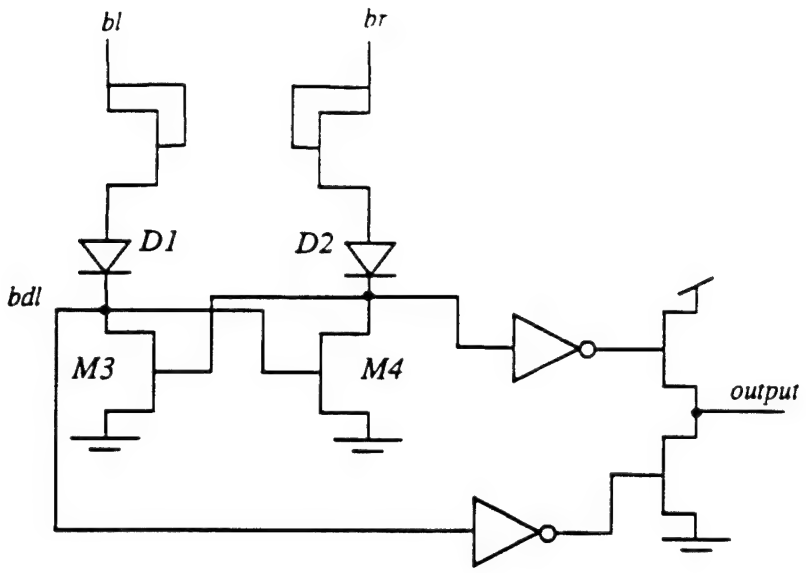


Fig. 3.19: New sense-amplifier schematic.

access times across process. The first approach to solving this problem was to develop a new sense-amplifier. A new sense-amplifier topology, shown in Fig. 3.19, is constructed by merging the input stage with the level conversion stage of the original sense-amplifier shown in Fig. 3.17. While this simplified the circuitry, it was not readily apparent how the new design compared with the original design. To facilitate the comparison, a software tool was developed to characterize the speed and process tolerance of a given sense-amplifier topology. This tool determines sense-amplifier transistor sizes that will produce a circuit that is functional over as large a process space as possible, while optimizing the speed of the sense-amplifier across process.

The flow chart for this methodology is shown in Fig. 3.20. The tool uses a SPICE deck of the circuit topology in which the sense-amplifier transistor sizes are parameterized. A discrete range of sizes for each of the variable-sized transistors is specified as an input. The user must also define the process space. To maximize the tolerance of the resultant sense-amplifier, a rectangular process space, rather than a tracking process space as described in Fig. 3.3, is assumed.

For each permutation of transistor size found from the range of specified sizes, the circuit is simulated at all four corners of a process rectangle centered around the nominal enhancement and depletion thresholds. The size of this rectangle is increased until a corner simulation results in

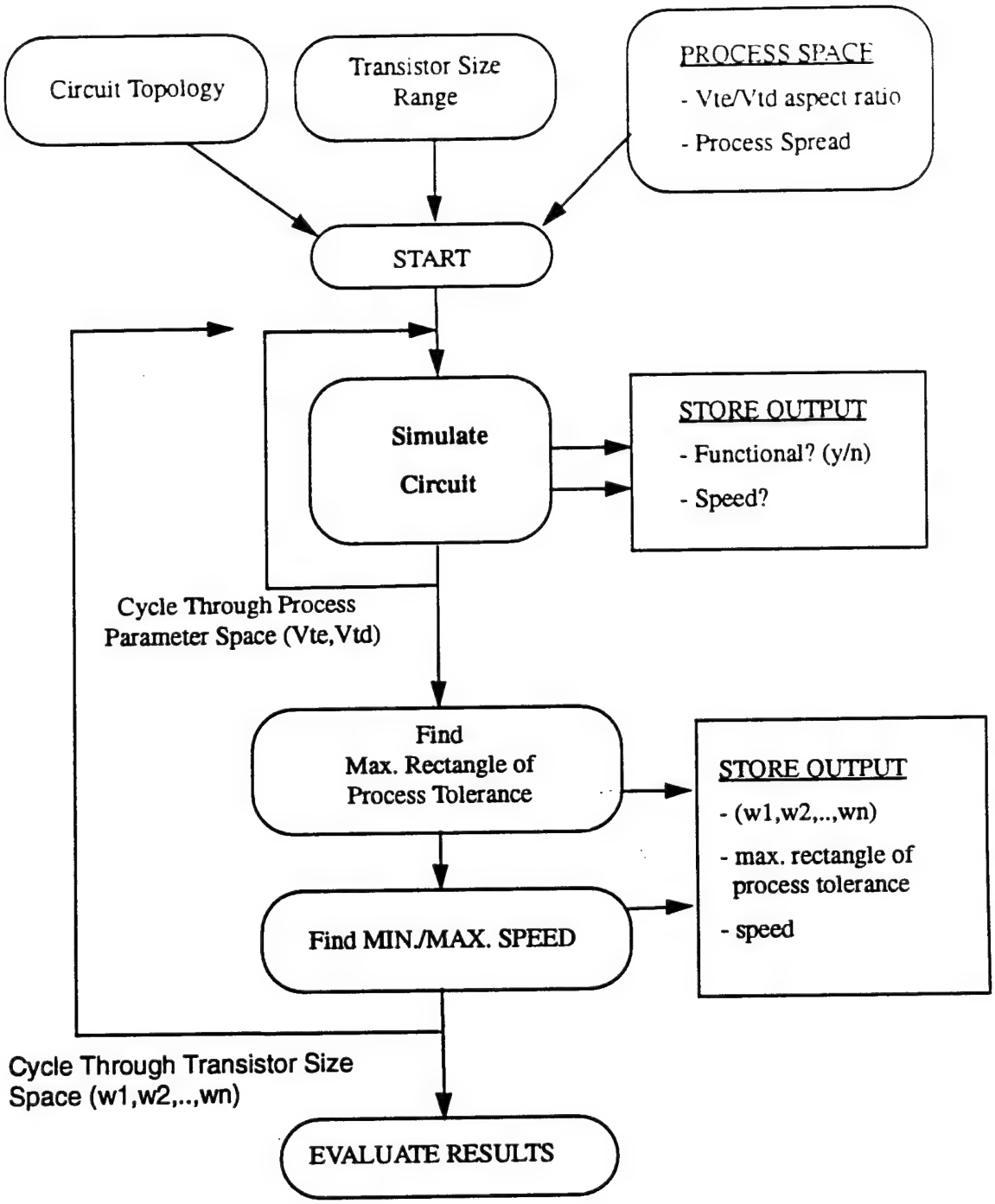


Fig. 3.20: Flow chart for process tolerant sense-amplifier design.

a non-functional circuit. Once a non-functional process corner is found, the maximum rectangle over which the sense-amplifier is functional, and the maximum value of access time over the functional process space is recorded. When the simulations are completed, these results are evaluated to choose an appropriate set of transistor sizes.

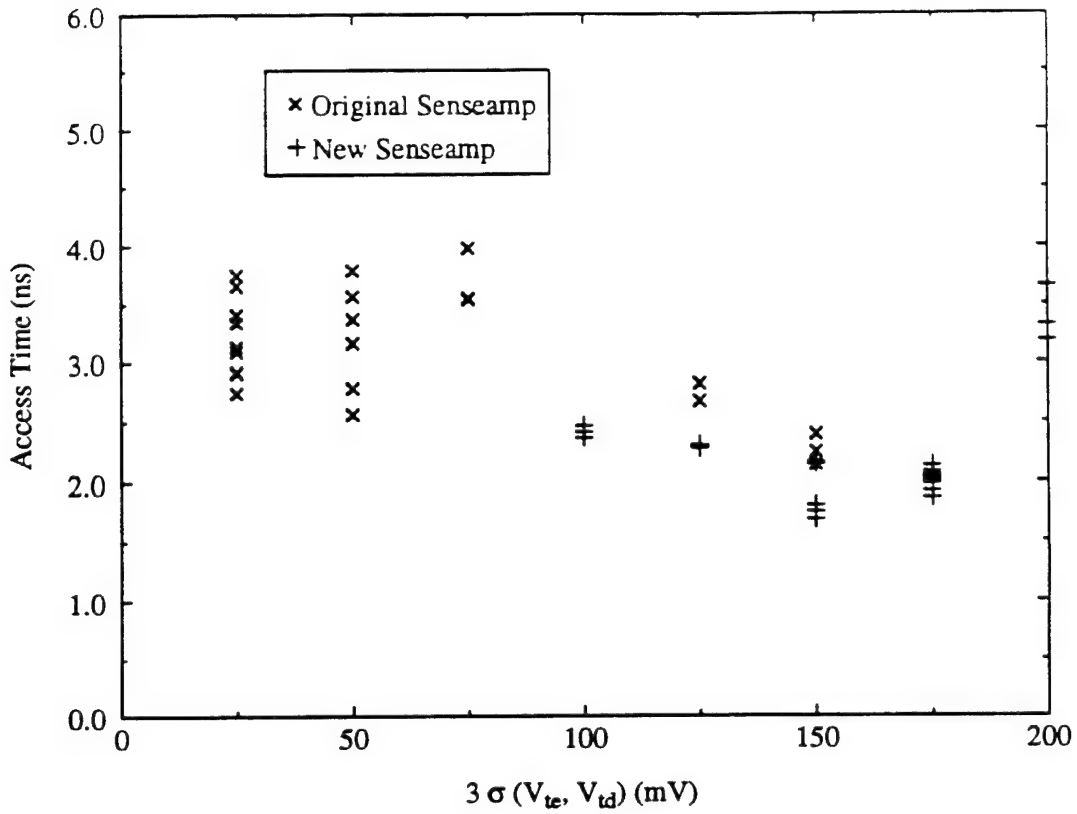


Fig. 3.21: Comparison of the speed and process tolerance of the two sense-amplifiers using the sense-amplifier characterization and transistor size selection algorithm.

This methodology was applied to the new and to the old sense-amplifier designs to compare their merits. The comparison was performed assuming a square process space, in 3σ process size increments of 25mV. The results are shown in Fig. 3.21.

This chart shows that for the old sense-amplifier design, process tolerant transistor sizes could have been found that offered 3σ (Vte, Vtd)=150mV. The chart also shows that the new sense-amplifier can achieve somewhat larger process tolerance with smaller access times than the old sense-amplifier. The new sense-amplifier achieved smaller delays because of the smaller parasitic capacitances. The new sense-amplifier shows larger process tolerance for a larger range of transistor size sets because it has fewer level-conversion stages.

3.4 Conclusions

The design of process-tolerant circuits requires an understanding of circuit failures in the presence of process variations. This chapter has presented a study of the design implications of

threshold voltage variations on GaAs DCFL circuits, high-drive buffers with feedback, and sense-amplifiers.

DCFL circuits were shown to fail at high temperatures, large fan-in, and low β ratios. A method for calculating the parametric yield of circuits in the presence of process variations was presented. This yield formulation can be applied to optimize target enhancement and depletion transistor thresholds for achieving high circuit yield, to determine the process parameter control needed to achieve desired yield for circuits of given integration levels, and make the necessary tradeoff in selecting β.

The effect of local and global process variations on various types of super-buffers that use feedback was also studied. Unlike DCFL circuits, super-buffers with feedback require careful control of the high-voltage noise margin. Feedback-FET logic buffers were found to be more tolerant than squeeze gates of both local and global process variations. The Feedback-FET logic buffer and squirt buffer exhibit equivalent tolerance to process variations; the FFL gate consumes less power than the squirt buffer for the same high-voltage noise margin. The design of FFL gates presents a trade-off between noise-margin and power-dissipation. An automatic sizing routine developed for these buffers must recognize this trade-off to ensure adequate noise-margins and to minimize unnecessary power dissipation.

Sense-amplifiers are made of stages that perform level-shifting and signal amplification. Each stage needs to be tolerant of variations in the common-mode voltages of the previous stage. In this chapter, an algorithm was presented for simultaneously optimizing the speed and tolerance to process variations. This methodology was used to compare and optimize two sense-amplifier designs. The more robust design was used in a 5-port register file, and exhibited much higher yield than the original design used in the asynchronous SRAM.

In the following chapters, design methodologies are presented which build on the design principles discussed in this chapter.

CHAPTER IV

POWER RAIL LOGIC: A LOW POWER LOGIC STYLE

In this chapter we describe a new logic style, called power rail logic (PRL). This new logic style is designed to be compatible with DCFL. Circuits are presented which reduce the power dissipation of digital GaAs logic functions from 10 to 40%. The logic gates that will be presented include an AND-OR circuit, a datapath multiplexor, a datapath latch, a datapath flip-flop, a datapath mux-latch-buffer, and an exclusive-OR gate. A 32-bit DCFL and a 32-bit PRL barrel shifter were designed, fabricated and tested. The PRL barrel shifter behaved correctly while achieving a 34% lower power-delay product than the DCFL barrel shifter, proving the viability of this new logic style.

4.1 Introduction

Two major circuit topologies are commonly used in large GaAs designs. The first of these is DCFL inverters and NOR gates. Large MESFET source resistances, back-gating, and low noise margins make it difficult to achieve high yield in GaAs designs which use NAND gates and pass transistor logic, which are thus discouraged. Although a complete set of boolean logic functions can be implemented with only NOR gates, shorter critical paths through control logic blocks can be achieved in technologies such as CMOS that also directly implement NANDs and complex gates such as XORs, and pass transistor logic. DCFL is preferred to other logic families such as buffered FET logic and source coupled FET logic because of its low power-speed product and low transistor count.

The second prominent circuit topology found in GaAs designs is feedback FET logic, which uses super buffers with feedback for efficient drive of large capacitive loads. The process

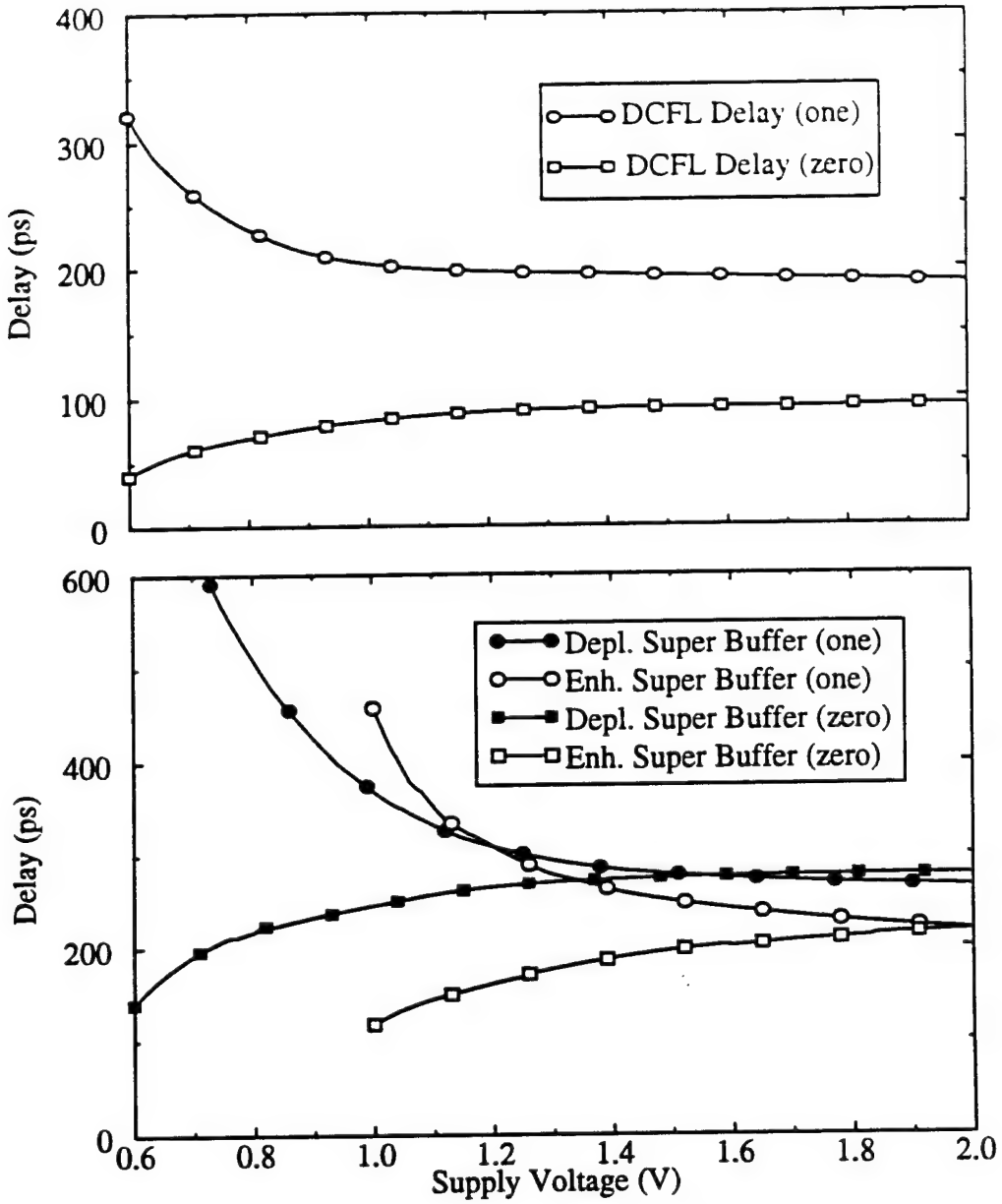


Fig. 4.1: Sensitivity of propagation delays to supply voltage for DCFL, enhancement and depletion super-buffers.

tolerance of this type of buffer was studied in Chapter 3.

Fig. 4.1 shows the dependence of propagation delays on supply voltage for a DCFL inverter, an FFL super-buffer with an enhancement pullup, and an FFL super-buffer with a depletion pullup. All three gates display functionality with supplies as low as 1.0V. Depletion FFL and DCFL gates exhibit functionality with power supplies as low as 0.6V. However, as the supply voltage drops below 1.4V, both types of FFL buffers show significant reductions in performance. VLSI chips such as Aurora I and II, which draw substantial amounts of current, can

exhibit power rail IR drops on the order of 0.5V from the pad ring to circuitry within the chip. Since the use of efficient, high-drive super-buffers is essential in VLSI DCFL, large, high-performance chips will continue to require power supply voltages of 1.4V plus the maximum IR drop in the power distribution network. Methods for reducing the IR drops, each of which has associated costs, include power distribution using pads in the center of the chip (as in area-interconnect MCM technology) and dedication of a metal layer for a Vcc mesh or plane.

Continued progress in CMOS and BiCMOS technologies has led to chip level performance that is comparable to that of large GaAs circuits. For GaAs to be competitive, it will need to show better delay and power dissipation results than are possible with existing circuit structures. Several circuit styles have been proposed which offer high performance at a low power dissipation [Hay85],[Gab87],[Tse87]. These designs, however, require as many as three to four power supplies which is very costly from a systems perspective.

Unless a new logic style completely eliminates the need for DCFL, the input and output levels should be compatible with DCFL circuitry. Finally, as emphasized by some of the results of Chapter 3, any new logic style must prove to be tolerant of supply voltage drops and process spreads.

A clever low-power logic family for GaAs called two-phase dynamic FET logic (TDFL)[Las93] has been presented. This new logic family is a very low power design style which offers NOT, NAND, NOR, AOI and XOR logic functions. It can achieve 88nW/MHz/gate as compared to 0.5μW/gate for 0.6μm DCFL, 5μW/MHz/gate for 3.3V 0.8μm CMOS and 8μW/MHz/gate for 5V BiCMOS. Logic gates in TDFL have been verified with power supplies between 1.0V and 2.0V. This logic family also has input and output signal levels that are compatible with DCFL, but it requires an extensive distribution of clocks that swing between 0V and -1.2V. The additional power supply must be routed throughout the chip. Because it is at a lower voltage than many of the transistor sources, this supply can cause backgating, which has somewhat unpredictable effects. This design style has not yet achieved high yields in large circuits.

In the next few sections we present a new logic style called power rail logic, which offers several advantages over straight DCFL circuits. While its power savings are not as great as those promised by TDFL, it does offer logic gates with lower power and lower power-delay products

than DCFL. This is achieved with a clean interface to DCFL circuits, requiring no additional power supplies and maintaining compatibility with DCFL signal levels.

4.2 A Methodology for Circuit Design, Characterization and Evaluation

In this section we present a methodology for designing, characterizing and evaluating circuits that has been applied to the evaluation of the new PRL circuits presented in the next sections.

Cell area, noise margins, process tolerance, supply voltage tolerance, speed and power dissipation for a particular load and switching frequency are figures of merit used for comparing circuits that compute the same logical function. In Chapter 3, the main objective in sense-amplifier design was to obtain a fast, process tolerant circuit. The methodology used did not consider parameters such as power dissipation or power-delay product.

For a given circuit topology, a designer can use intuition to size transistors. While intuition can guide the designer through the n-dimensional space of transistor sizes, a more extensive search is needed to optimize the transistor sizes. When comparing logic styles, the user must consider circuits having properly sized transistors. Only those sets of transistor sizes that produce functional circuits across process variations, temperature and supply voltage can be used in such comparisons.

Figure 4.2 is a flowchart for the methodology we have developed for circuit design and evaluation. First, given a circuit, certain transistors (such as input gates and output loads) are fixed in size. Next, ranges and ratios of size are specified for the remaining transistors. We will use the notation W(Mn) to refer to the variable size of transistor n. The transistor length is usually assumed to be the minimum length allowed by the process.

A parameter input file is constructed which contains the range of transistor sizes and ratios for each of the transistors and transistor ratios to be varied. For example,

W1(M1),W2(M1),W3(M1)

W1(M2),W2(M2),W3(M2),W4(M2)

W1(M3),W2(M3),W3(M3)

W1(M4),W2(M4)

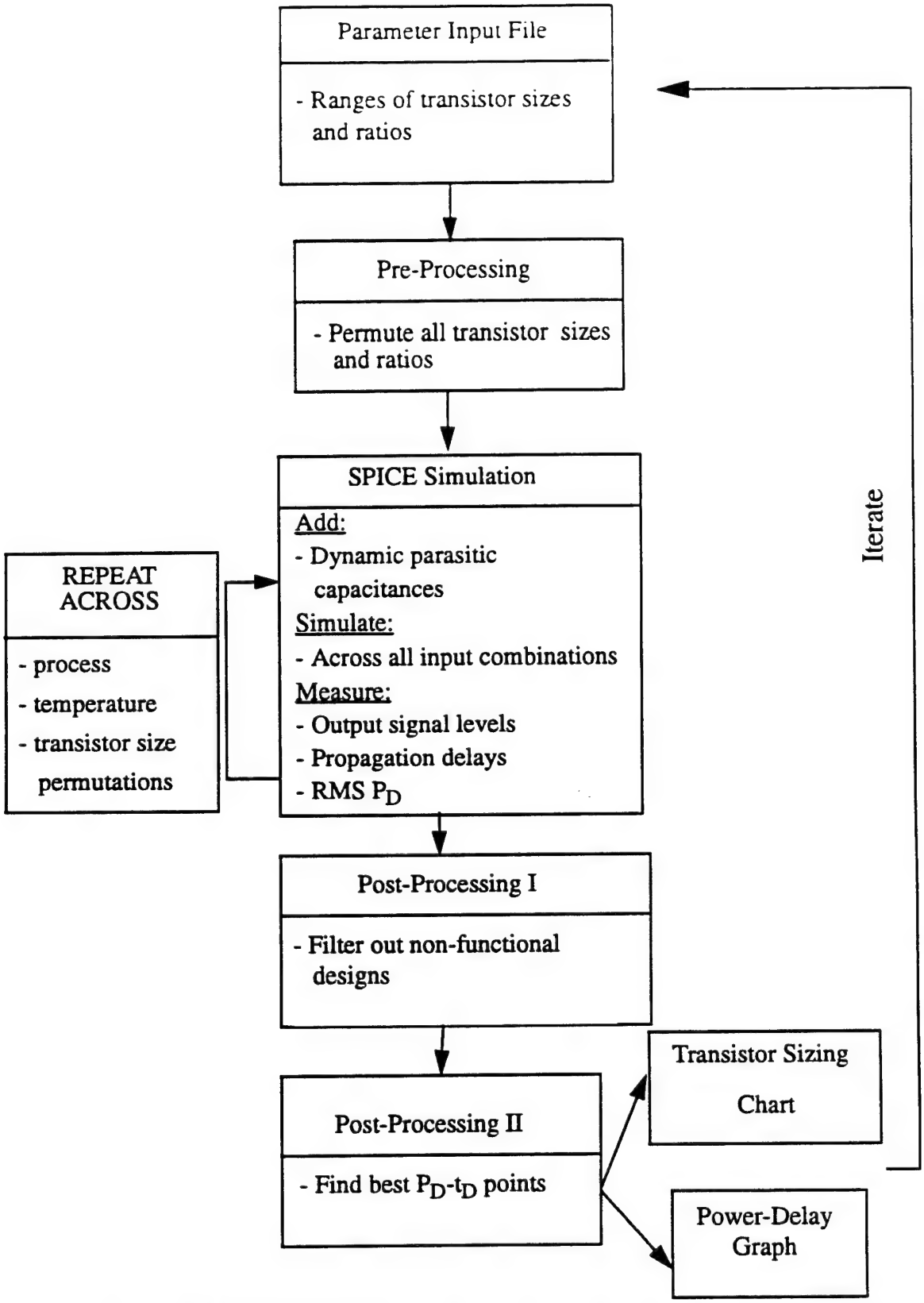


Figure 4.2: Methodology for process tolerant circuit design and comparison.

would represent an input parameter file for a circuit having four variable-sized transistors. In this example, the designer wishes to consider at the performance of a circuit using three different sizes for transistor M1, four different sizes for M2, three for M3 and two for M4.

A pre-processor then permutes all of the sizes of all of the variable-sized transistors from the parameter input file in a format compatible with a SPICE deck. Constraints sometimes need to be placed on transistor sizes. For instance, if we know that the circuit will only be functional for W(M1)<W(M2), it would be computationally wasteful to simulate cases where W(M1)>W(M2). Such constraints can easily be coded into the pre-processor. For this example, without any constraints, the pre-processor will generate simulations for 3x4x3x2 = 72 sets of transistor sizes to be simulated.

A SPICE netlist is created for the circuit topology. In the SPICE deck, parasitic capacitances are dynamically calculated and attached to each node based upon the transistor dimensions. For instance, the parasitic capacitance on a node x may be a function of W(Mx), and would be calculated as

$$C_n = \alpha + \beta \cdot W(M_x).$$

Depending on the layout methodology (datapath vs. custom layout), different proportionality constants β (with units of fF/μm) can be used for this capacitance estimation.

The SPICE simulation is set up so that it exercises the circuit across all possible input combinations, while observing all output transitions. Measure statements are used to record low-to-high and high-to-low propagation delays, output signal levels, and the circuit RMS power dissipation across all input combinations. Since some circuits include significant transient power demands, the circuit is stimulated at its operating frequency. For the circuits described here, the simulation frequency was 250MHz.

The circuit is simulated once for each permutation of transistor sizes generated by the pre-processor at every process corner. Based upon the process spread information provided by vendors (Vitesse and Cray in this case), a 3-point (3σ slow-slow, typical-typical, 3σ fast-fast) process model is used. Optionally, this analysis may be performed across temperature and supply voltage. For this example, the number of simulations run would be

$$Simulations = 3\frac{simulations}{set-of-sizes} \times 72 \cdot set-of-sizes = 216.$$

The temperature and supply voltage variation analysis would then be performed only on the set of sizes that passed the process variation tests.

After all the simulations and measurements are completed, the SPICE output will consist of output voltages measured at specified times, propagation delays, and power dissipations measured for each permutation of transistor sizes at each process corner. This output is fed into a two stage post processor. In the first stage, the output voltage measurements are used to determine whether or not a given set of sizes yields a logically functional circuit that meets output voltage noise margin constraints over process variation. In this stage, all sets of transistor sizes that produced non-functional circuits are discarded.

The second stage takes the remaining data and sorts the solution set by power-delay product. The power-delay product used for comparison can be chosen from the typical process corner or the process corner that gives the largest power-delay product. This list of vectors of the form {power-delay product, power, delay, transistor sizes} is then sorted by power-delay products and applied to a sifting function. This function selects only those sets of transistor sizes that exhibit the smallest power-delay products and lie on a monotonically descreasing P_D-t_D curve. As an example, the circles in Fig. 4.3 show a set of data which was produced by a circuit and passed through the first stage of the post processor. Each data point produced by the first stage of the post-processor represents a set of transistor sizes for the circuit that exhibited functionality across all process corners. The data points that were selected by the sifting function in the second stage of the post processor are also shown as stars in this graph.

After the sifting function is applied, the post-processor generates a transistor sizing chart file which contains transistor sizes, power dissipations and propagation delays. This information can be used as an aid to circuit design. The second file contains the power and delay points from the first file for plotting a characteristic P_D-t_D curve. This characteristic curve can then be used for comparison with other circuits that compute the same function. Since the results of this analysis are limited by the initial choice of ranges of parameters, iterations of this procedure are performed until the designer is confident that the search has been extensive enough to find the best power-

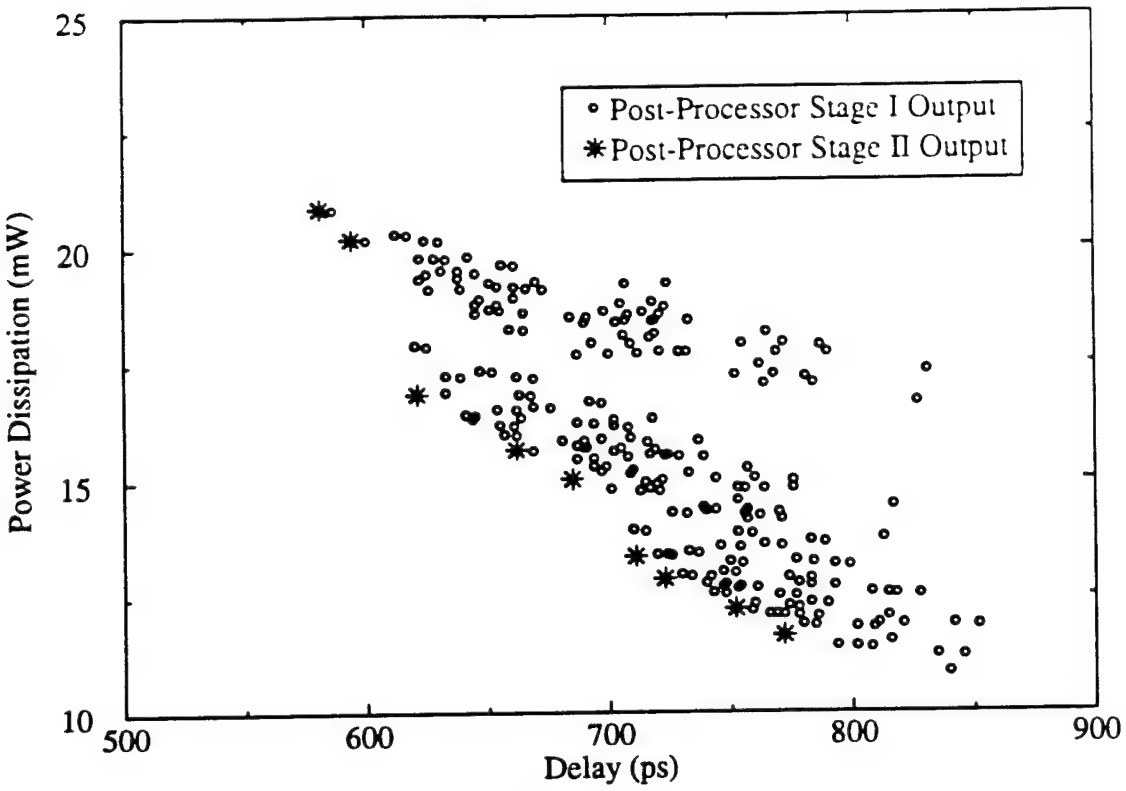


Figure 4.3: Sample output data from post-processor stages I and II.

delay products for the circuit. This technique has been applied to PRL and DCFL circuits, to assure that the comparison of these logic families is accurate. On average, approximately ten thousand sets of transistor sizes were used to evaluate each circuit.

4.3 Power Rail Logic Circuits

First we present the PRL inverter, the basic building block from which all other PRL gates are constructed. This gate is shown along with its schematic representation in Fig. 4.4(a). Topologically, this gate is identical to a DCFL inverter, with input signal A used to control the power rail of the inverter. The logic levels used for input B are the DCFL levels of 0.1V and 0.6V. When input A is brought low (to ground) the depletion load transistor remains on and ensures that the output is low. When input A is raised to a voltage above approximately 0.6V the logic gate behaves like an inverter, producing $\overline{B}$ on the output.

To achieve the shortest delay from B to the output, input A must be raised to at least 0.9V. The resulting function computed by this gate is the and-invert function $A\overline{B}$. This function can be

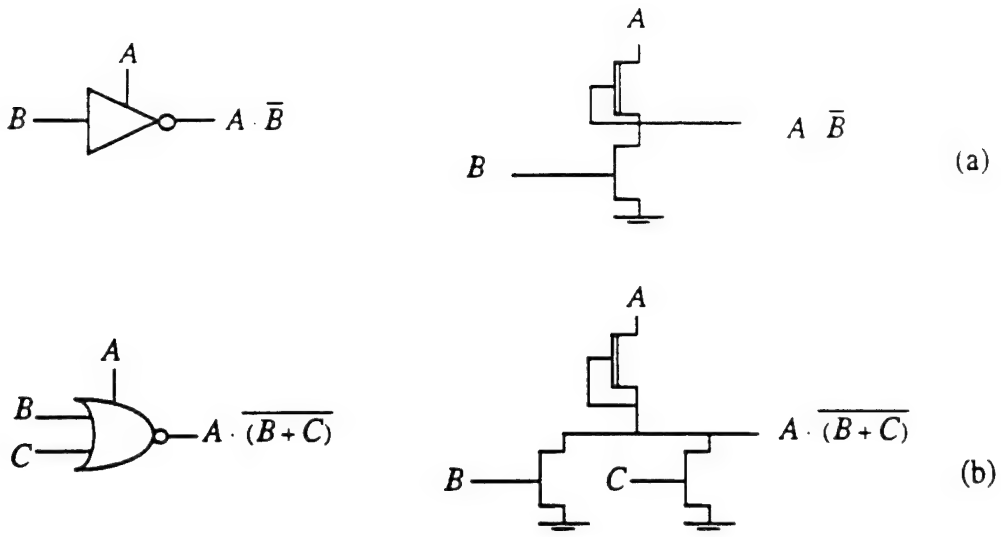


Figure 4.4: Basic power rail logic (PRL) gates.
(a) PRL inverter (b) PRL NOR-gate.

easily extended using additional pulldown transistors, as in Fig. 4.4(b), to produce the AND-NOR function.

Among the most common datapath elements used in digital applications are multiplexors, latches and flip-flops. Power rail logic can be applied to all of these circuit structures to reduce power dissipation with little or no impact on performance. Other functions such as the XOR can also be implemented using PRL gates. The next five sections describe PRL circuits, illustrating different aspects of this new logic style.

4.4 A PRL Four-Input Multiplexor Datapath

Most digital circuits can be categorized as datapaths, standard cell control logic blocks, memory, or I/O cells. Datapaths are regular, modular circuit blocks that use n identical circuits to implement an operation on n bits of data in a bit-wise manner. Data flows in one direction through a datapath while control signals are routed in the orthogonal direction, as shown in Fig. 4.5. In the Cascade Design Automation software that we use for digital IC design, a column driver, associated with a particular datapath column, is used to supply control signals to the datapath module.

Power rail logic is particularly well-suited to implement datapath logic; the four-input multiplexor will be described as an example. A DCFL four-input datapath multiplexor is shown in

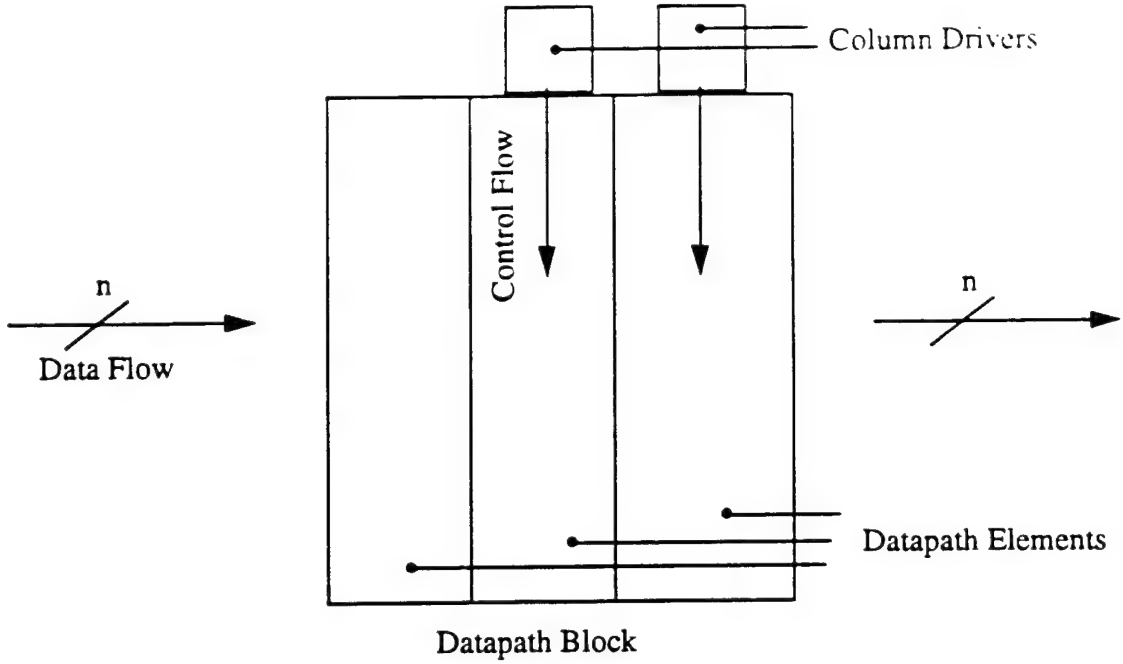


Figure 4.5: Block diagram of the datapath paradigm.

Fig. 4.6. The input stage consists of a set of four two-input NOR gates, each having a decoded select input and a data input. At any given time, three of the four select lines are driven high, and the fourth line is driven low. This sensitizes a path for the fourth data input to pass through the multiplexor. Each of the NOR gates typically draws 100-150μA of current, dissipating 200-250μW of power with a 2-V supply for a total of 1-1.25mW per bit. Since the only purpose of three of the five gates is to produce a zero output, they are drawing supply current needlessly. The column drivers that drive the four select lines are each feedback-FET logic buffers, which themselves draw approximately 1.1mA. The schematic for the column driver circuitry is shown in Fig. 4.6.

Fig. 4.7 shows a power rail logic four-input multiplexor with associated column drive circuitry. In a DCFL mux, the select lines are associated with the gate terminals of pulldown transistors on the input NOR gates. In the PRL mux, the select lines are instead connected to the power rails of the input inverters. To select a given input, the power rail control signal connected to this input is raised instead of lowered, allowing the input to be inverted and passed to the output NOR gate. The remaining three power rail control lines are brought to ground. Since the control lines are connected through depletion transistors to the outputs of the input inverters, the

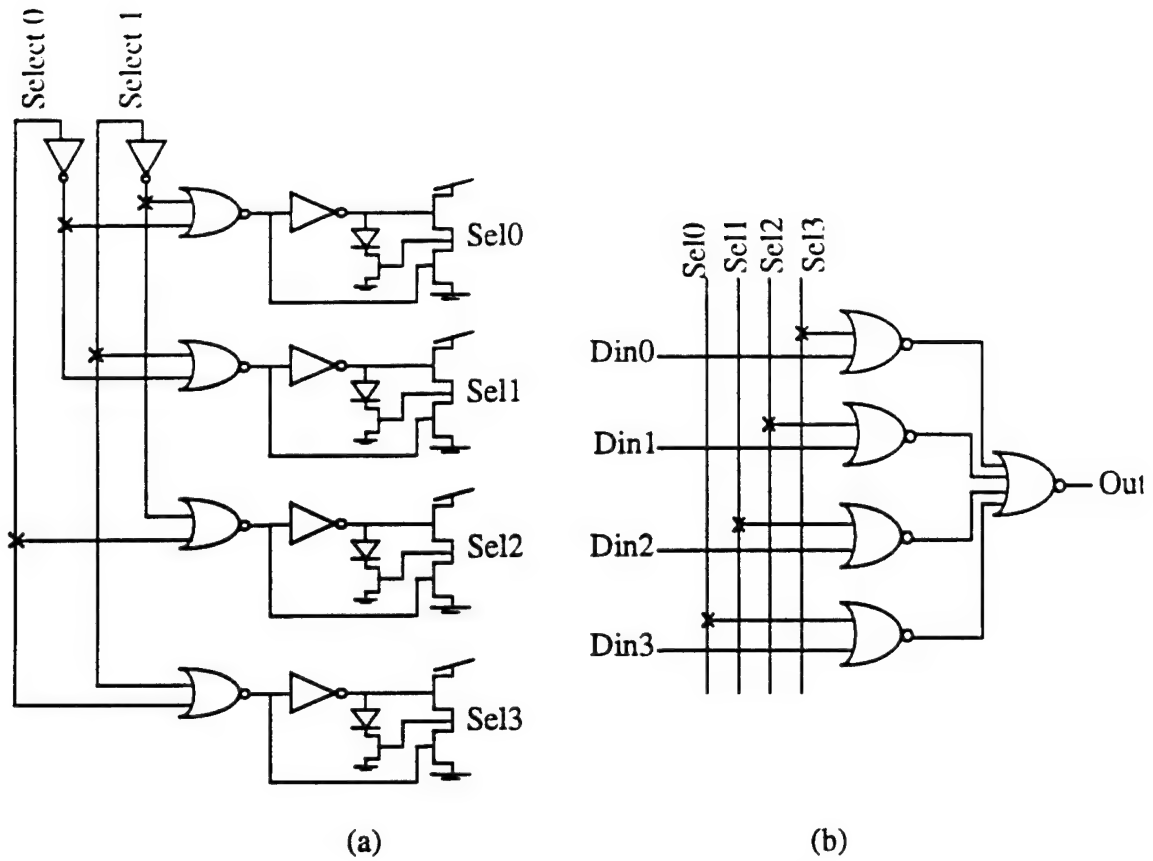


Figure 4.6: A DCFL four-input datapath multiplexor.
(a) column driver, (b) datapath cell.

remaining three inputs to the output NOR gate are brought to ground, sensitizing a path from the selected input to the output of the multiplexor.

This configuration has a number of advantages over the conventional DCFL mux. Since only one input inverter will be selected at a time, only one of the four input NOR gates will draw any current from the power supply. This can lead to substantial power savings. The new multiplexor is also more compact than the DCFL layout. The transistor count for a PRL four-input mux is 13, compared to 17 for the DCFL mux. Since only the output NOR-gate needs to be connected directly to the power supply, the amount of Vcc routing within the cell is reduced.

The added cost associated with this multiplexor is in delay, since the control signal must swing by at least 0.8V to ensure adequate operation[1] instead of the 0.6V swing associated with the

[1] A discussion on the effect of the power rail control line high voltage on the operation of a two-stage PRL gate is given in section 4.5.2.

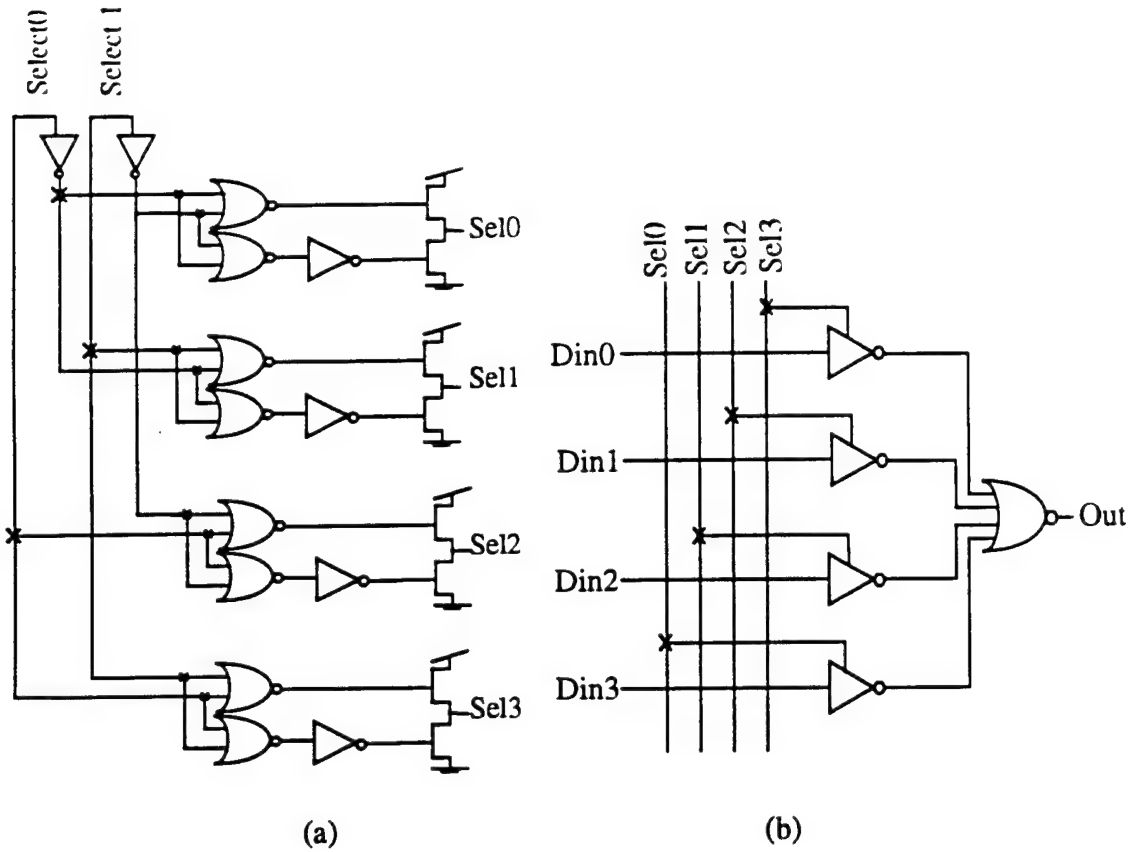


Figure 4.7: A PRL four-input datapath multiplexor.
(a) column driver, (b) datapath cell.

DCFL mux select line. This cost can easily be mitigated by spending some of the power saved in the datapath multiplexor cells to provide higher drive for the power rail control lines.

The methodology described in section 4.2 has been applied to compare the power-delay characteristics of a 32-bit PRL four-input mux datapath and a 32-bit DCFL four-input mux datapath. The simulations used capacitances extracted from layouts. Fig. 4.8 shows the power-delay curves generated for these multiplexor circuits operated at 2V. The delay is measured from the input select lines of the column driver through the column drive decoder, through the column drivers, through the furthest load multiplexor to its output. The power and delay points used for comparison are those for transistor sizes that achieved the best overall power-delay products, using the worst power-delay product measured across process variations. The load of the output multiplexor was fixed at 100fF plus a 0.8 x 20μm diode. This load is representative of driving across two to three datapath elements with a fanout of 3 minimum-sized inverter loads. Since three of the five gates in the PRL mux are off, we would expect the upper bound for power savings

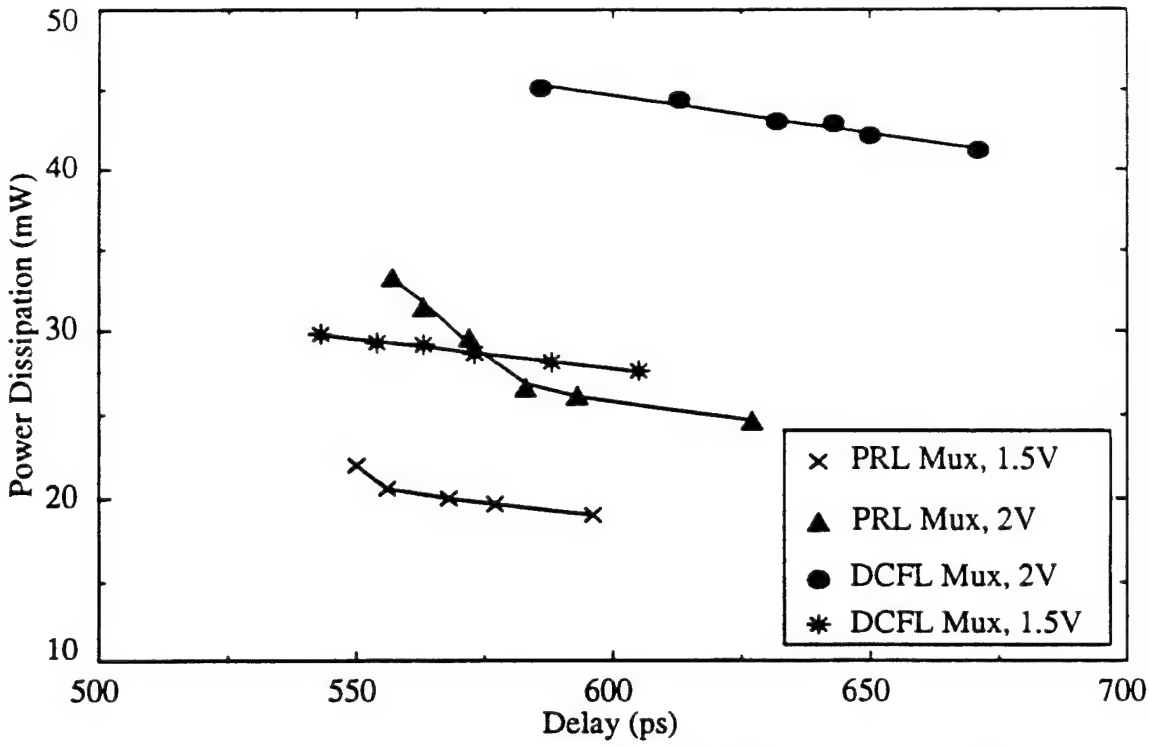


Figure 4.8: Power-delay curves for a 32-bit DCFL and PRL four-input mux datapaths.

to be 60%. The real savings are less than this since the output NOR needs to be sized up to drive the load capacitance efficiently; power savings are also reduced due to the power dissipated in the column drivers. Despite these factors, the PRL datapath mux circuit achieved a 30% savings in power over the DCFL datapath.

4.4.1 Supply Voltage Tolerance

The circuit characterization methodology was again applied to the DCFL and PRL mux circuits assuming a 1.5V supply. The results of this analysis are also plotted in Fig. 4.8. At a supply voltage of 2.0V the PRL datapath mux exhibits power dissipation and power delay products comparable to that of a DCFL datapath mux optimized for a 1.5V supply. In designing the PRL mux circuitry to operate at lower supply voltages, one must be certain that the power rail select lines achieve a high enough voltage to prevent degraded performance in the PRL inverters being driven. This larger swing is achieved by increasing the size of the output pullup transistor of the column driver shown in Fig. 4.7(a). Fig. 4.9 shows the effect of supply voltage on PRL propagation delays for a mux designed to operate at 2.0V and one designed to operate at 1.5V. By

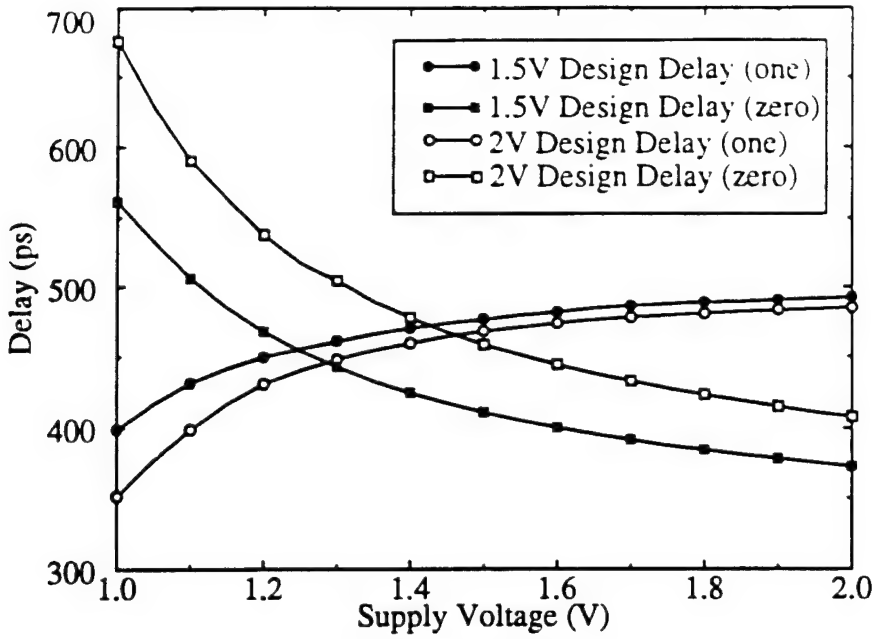


Figure 4.9: Effect of supply voltage on PRL datapath propagation delays.

using larger output driver transistors on the column driver (in this case 60µm vs. 50µm), the 1.5V design does not show performance degradation until Vcc=1.1V, as compared to 1.4V for the 2.0V design. The 1.5V design compares favorably in terms of supply voltage tolerance with the DCFL datapath mux shown in Fig. 4.10, which starts showing performance degradation at 1.05V.

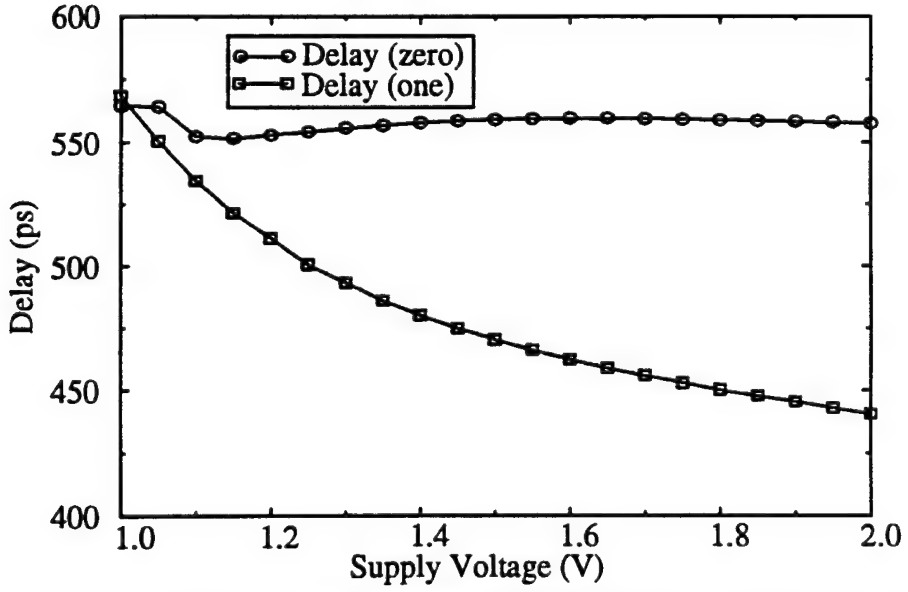


Figure 4.10: Effect of supply voltage on DCFL datapath propagation delays.

4.4.2 Components of Power Dissipation

The components of power dissipation for both datapath implementations are given in Table 4.1. This data was taken for equivalent performance DCFL and PRL implementations at the typical-typical process corner. The input stage of the PRL implementation is not directly connected to the supply voltage and is therefore not considered to contribute to overall power dissipation. Power dissipated in the PRL input stage is accounted for in the column drivers, which are directly connected to Vcc. The major component of power in the DCFL datapath (almost 45%) is dissipated in the input multiplexor stage. In this simulation, both multiplexor types are driving equivalent loads. As expected, for them to achieve similar delays, output stages of similar size are required, leading to equal power dissipations. To drive the moderately large load (100fF and 20μm gate width), the buffers in this example were made large; they drew the largest component of power in the PRL mux and the second largest component in the DCFL mux.

The PRL gate has only one of its four super-buffer column drivers output on at a time, dissipating 6.5mW at 2V; the three other column driver outputs are low, dissipating a total of 5.9mW. In comparison, at any given time, three of the DCFL drivers will be driving logic highs, dissipating a total of 5.8mW, while one will be driving a logic low, dissipating 1.4mW.

Overall the PRL datapath multiplexors exhibited a significant power savings for equivalent delay, showing for this data point a 31% savings at 2.0V and a 25% savings at 1.5V. The PRL and DCFL implementations exhibit similar sensitivities to process variation. The power dissipations across process corners at 2.0V vary from 31mW at the slow-slow corner to 42mW at the typical-typical corner to 54mW at the fast-fast corner for the PRL mux, and 45mW, 57mW and 71mW for the DCFL mux at the same corners, for savings of 31%, 26% and 24%. The delays

Table 4.1: Power dissipation components for the DCFL and PRL datapath multiplexors

Circuit	**PRL 2.0V**	**DCFL 2.0V**	**PRL 1.5V**	**DCFL 1.5V**
Mux Input Stage	0.0mW	20.3mW	0.0mW	13.1mW
Mux Output Stage	18.5mW	17.2mW	13.5mW	11.2mW
On Column Drivers	6.5mW	5.8mW	4.0mW	3.6mW
Off Column Drivers	5.9mW	1.4mW	4.1mW	1.0mW
Col. Driver Decode	0.3mW	0.3mW	0.2mW	0.2mW
Total	**31.2mW**	**45.0mW**	**21.8mW**	**29.1mW**

for the two circuits also follow this trend across these process corners.

4.4.3 Implementation Considerations

Practical considerations must be made when implementing PRL circuitry. These include sizing the control line drivers adequately with respect to their loads, observing electromigration limits on the control lines, and ensuring that the voltage drops along the length of the power control line do not adversely affect circuit performance. For a DCFL multiplexor column driver, the control signal being driven is the gate of an enhancement MESFET. These inputs can be turned on with only 25μA each or about 1/4 the current needed to drive a PRL control line. Thus, electromigration limits and voltage drops are a greater concern for PRL circuits. In the physical design methodology we have adopted, control lines are routed in third layer metal, which has the lowest resistivity and highest electromigration limits of all of the routable metal layers. Consequently performance degradation caused by I-R drops is insignificant, and complying with electromigration limits is easy.

A problem that is commonly associated with alternative low power circuits is large transient demands in supply current. This is not a problem for the datapath multiplexor. Fig. 4.11 shows the supply current drawn by the DCFL and PRL 32-bit four-input multiplexor datapath modules as the select lines are being switched. Fig. 4.11(a) shows the transient supply current demand for the FFL column drivers used with the DCFL circuits. The PRL column driver demand is shown in 4.11(b) for comparison. Fig. 4.11(c) shows the total supply current demand for both implementations.

The PRL column drivers in this simulation are switching as much as 5mA each. Depending upon the data being passed through the multiplexor, this current may vary, as shown in the figure. When the PRL inverter is driving a logic ONE, V_{DS} of the depletion load is suppressed, leading to a smaller demand in supply current from the column driver. In the simulations, all of the inputs are alternately forced high and then low. Since one column driver is switching off at the same time that another is switching on, the net transient demand for supply current (Fig. 4.11c) for the PRL implementation is minimal. The feedback-FET logic column drivers have a high demand for supply current at the positive-going edge, and hence the DCFL datapath mux exhibits more

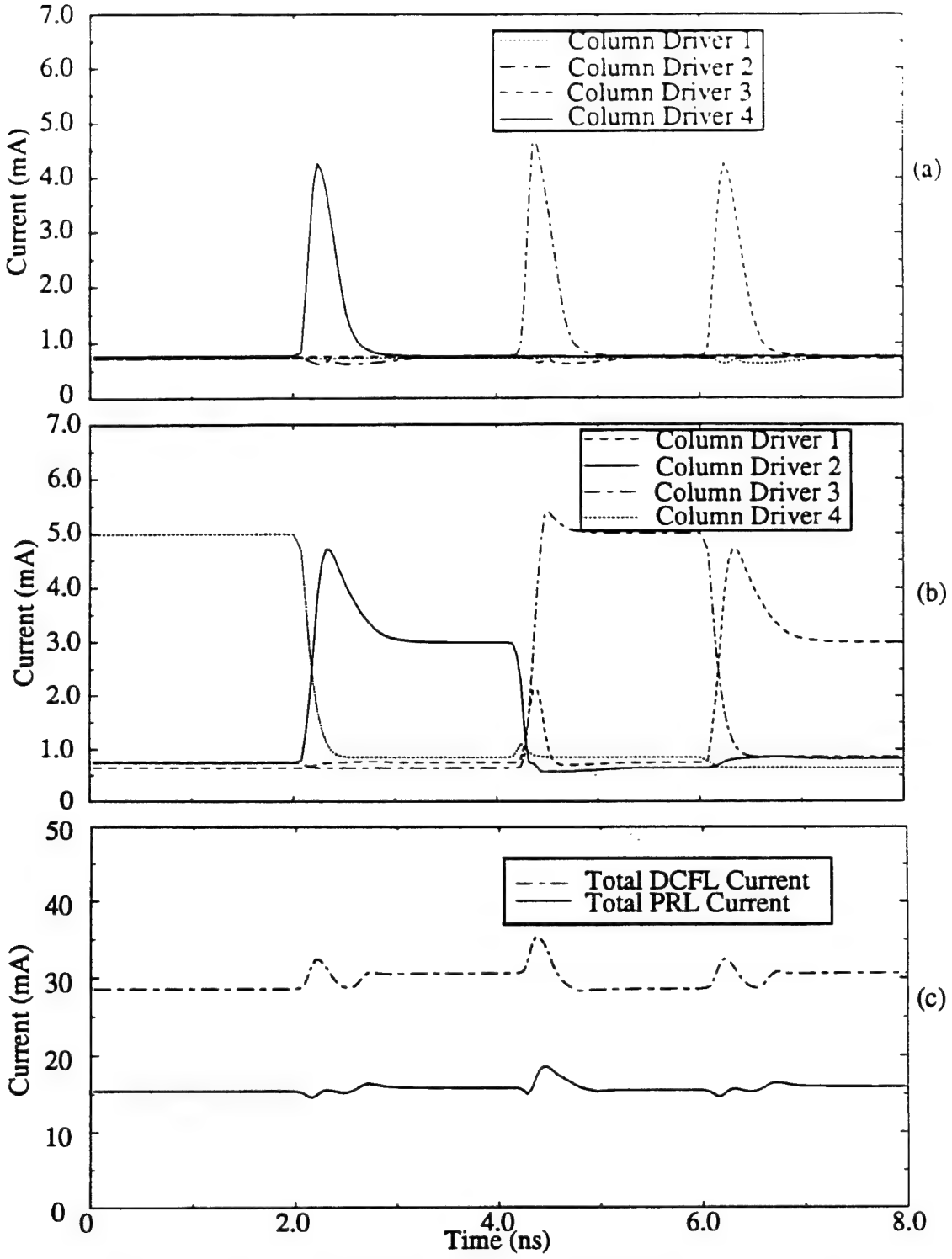


Figure 4.11: Transient supply current demands for the PRL and DCFL datapaths. (a) DCFL column driver demands; (b) PRL column driver demands; (c) total datapath demands.

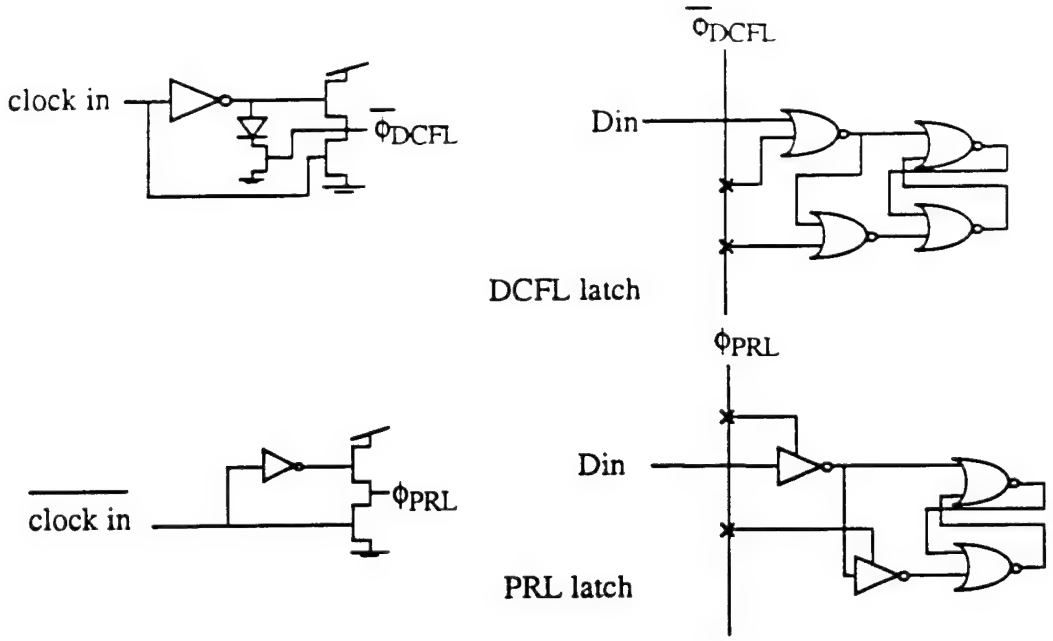


Figure 4.12: DCFL and PRL latch and column driver circuits.

current spiking than the PRL mux.

In other PRL circuits where only one select line is being switched at a time, such as the datapath latch which is described in the next section, the PRL datapath exhibits larger transient demands, similar to those of the equivalent DCFL datapath.

4.5 A PRL Latch

The major advantage of the PRL mux is that it can achieve substantial power savings (up to 40%) over DCFL by turning off gates that are not needed. Power rail logic circuits can also achieve smaller delays from control signal input to data-output than DCFL, as will be illustrated with the PRL latch presented in this section.

A PRL latch can be constructed from a DCFL latch in a manner similar to that by which the PRL mux was derived from the DCFL mux. These two types of latches and their associated column drivers are shown in Fig. 4.12. The clock input of the DCFL latch acts to pass Din and $\overline{\text{Din}}$ when $\overline{\phi}_{DCFL}$ is low, and latches the outputs by forcing the outputs of the two input NOR gates low when $\overline{\phi}_{DCFL}$ is high. This same function can be accomplished by using clock ϕ_{PRL} to control the power rail of the two input inverters. For this latch, driving ϕ_{PRL} high makes the latch transparent

while bringing ϕ_{PRL} low latches the outputs.

In the PRL latch there is an opportunity to save power whenever the outputs are latched by shutting off the input inverter gates. Most latches implemented in a pipelined system are transparent during one half of a clock cycle and the outputs are latched for the second half. By turning off one half of the gates one half of the time an upper bound of 25% power dissipation savings can be expected. If the latch is driving a large capacitive load, the output NOR gate must be made larger to minimize the total propagation delay. This increases power dissipation in the output stage, and reduces the percentage of total power that can be saved by shutting off the input stage during the latched mode.

A comparative study was performed on 32-bit DCFL and PRL datapath latches optimized to drive small and moderate size of capacitances of 20fF and 100fF[1]. The output NOR gates of both implementations and the input NOR gates/inverters were allowed to take on the same sets of sizes as each other. The column driver output transistors were allowed to assume the same sizes as long as electromigration limits were observed. The delays measured were from both the clock and data-inputs switching simultaneously to the data-output.

4.5.1 Moderately Loaded Datapath Latches

The manner in which transistors need to be sized for minimum power-delay products is somewhat different for the two circuits. The clock-input to data-output delay of the each datapath latch is very sensitive to the size of column driver output transistors. Fig. 4.13 shows a family of power-delay curves generated for both circuits using output column driver transistors of size 60μm, 120μm and 180μm.

For a given size of column driver, the PRL implementation exhibits a better power-delay curve than the DCFL implementation. To pitch match column-drivers to their datapath cells, and to minimize chip area, it is desirable to make the column drivers as small as possible. Output transistors on these drivers as wide as 120 and 180μm become excessively large given that the

---

1. For capacitive loads larger than 100fF a super-buffer with feedback would be more appropriate on the output stage.

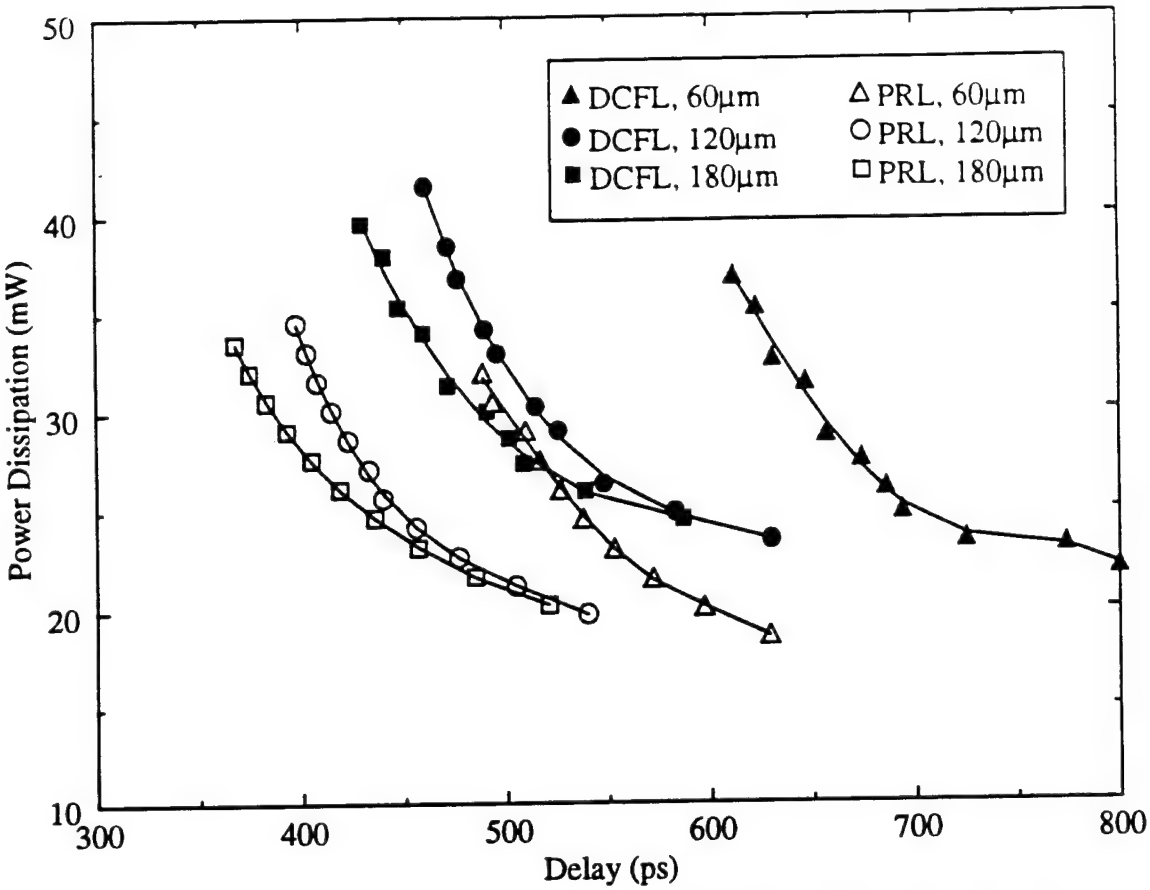


Figure 4.13: Power-delay curves for a moderately loaded DCFL and PRL latch for different sized column driver pullup transistors.

height of individual datapath cells is only on the order of 55µm. The 60µm wide column driver is the most practical of the ones studied. PRL gates inherently have a smaller delay from the control line input to the gate output than DCFL gates. Given this fact (to be shown below) and the very steep nature of the power-delay curves, we see effective power savings for the PRL latch that are much larger than the 25% "upper bound". For instance, a comparison of latches using 60µm column drivers shows that the PRL latch only requires 19.1mW at 630ps delay, whereas the DCFL latch requires 32.5mW to achieve the same delay. Because of its speed advantage, power savings of PRL in this case are thus 41%, which is much more than the expected upper bound.

The salient feature of these curves, however, is not that the moderately loaded PRL datapath latch can achieve lower power dissipations but that it can achieve smaller delays than the DCFL latch. In the next section we offer some insight into how PRL gates are sized and describe why power rail logic gates can achieve comparable and even smaller delays than DCFL.

4.5.2 Delays through a PRL Datapath

The PRL datapaths that have been presented thus far, including the multiplexor and the latch, share the same basic structure. A control signal, such as the select line for the multiplexor or the clock line for the latch, is used to either pass or block a data signal from flowing through a logic gate. When the control line is used to block the data signal, the output of the gate being controlled is forced low. When the control line is used to pass the data signal to the next level of logic, the data is inverted as it is passed. The second stage then inverts the signal again and drives the output of the circuit.

There are four basic delays associated with these circuits: the propagation delays for passing a ZERO or a ONE from data-input to data-output (with stable control signals), the delay from the input of the column driver to data-output when passing data to the output, and the delay from the input of the column driver to the output of the first stage of logic when blocking input data. Each of these delays will be described individually.

4.5.2.1 Data-Input to Data-Output Delays

The first delays described will be those from data-input to data-output with stable control signals. For both the DCFL and PRL gates these are propagation delays through two levels of logic, which might appear to be identical. When passing a logic low, the first stage inverts the signal and drives a logic high into the second stage. The first-stage delay is a function of the current drive available from the first stage, and is thus dependent not only on the size of the depletion load but also on how high the power rail control line is raised.

The control line high voltage can be set by sizing the enhancement pullup transistor in the column driver. Two factors cause a reduction in this voltage; I-R drops along power lines leading to the column driver and along the control lines. Simulation results show that the I-R drop from the column driver to the power-rail input of the gate farthest from the driver can be as much as 100mV. The second cause of reduced power rail control voltage is data-dependent. When most of the data-in signals are high, their associated inverter outputs are low, causing a strong coupling between the power rail control line and ground. Hence the power rail high voltage depends also on the number of data-input signals that are high. Fig. 4.14 shows the relationship between column

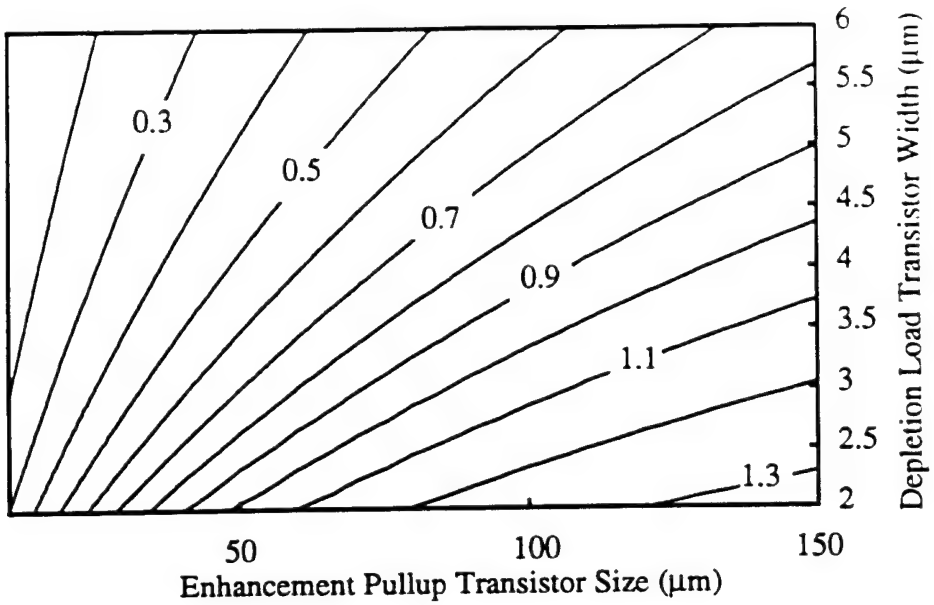


Figure 4.14: Power rail control voltage high level as a function of column driver and depletion load transistor sizes with a 2.0V supply.

driver pullup-transistor size, depletion load width and power rail high voltage, assuming a 2.0V supply and a 32-bit datapath. A typical minimum-sized depletion load transistor would have a width of 2µm. The voltage shown in this figure was measured at the furthest end from the column driver with all inputs high. The sensitivity of this voltage to the supply voltage can be appreciated with the aid of Fig. 4.15, which shows this same relationship with a supply voltage of 1.5V at the

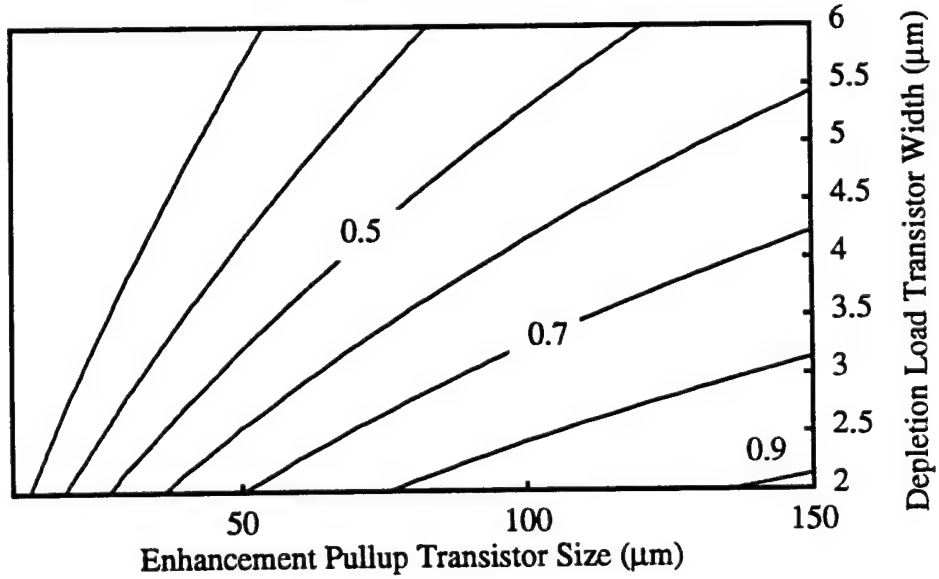


Figure 4.15: Power rail control voltage high level as a function of column driver and depletion load transistor sizes with a 1.5V supply.

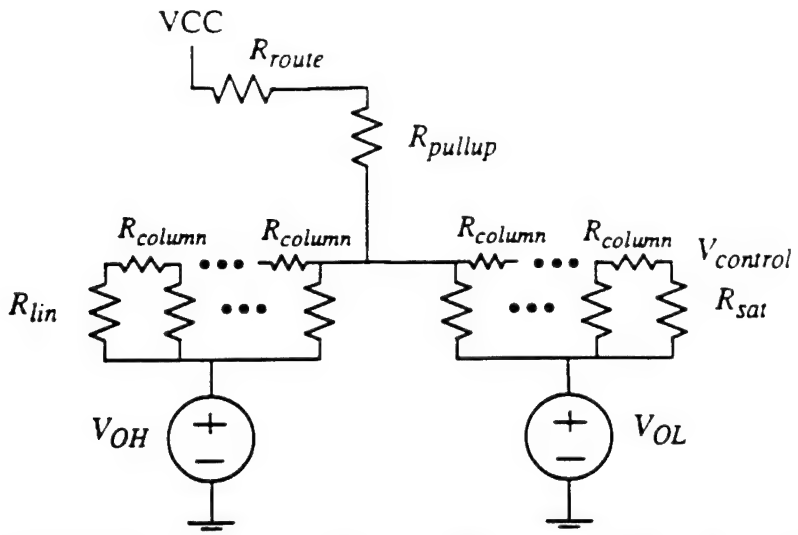


Figure 4.16: Approximate equivalent circuit of a PRL datapath module and a column driver.

column driver. These charts can be used to help select depletion load transistor and column driver sizes to achieve an acceptable control line high voltage. Fig. 4.15 shows, as expected, that much larger enhancement pullup transistors are required with a 1.5V supply to achieve the same power rail high voltage as with a 2.0V supply.

The data-dependent nature of the control line high voltage is illustrated in Fig. 4.16, an equivalent circuit of a PRL datapath and a column driver. R_{route} represents the resistance in the power line leading from off-chip Vcc to the column driver. R_{pullup} is the on-resistance of the pullup transistor in the column driver. The series resistance of each segment of the control line leading to the cells is given by R_{column}. R_{sat} and R_{lin} represent the resistances of the depletion load transistors which have their source nodes at output low and output high voltages respectively. R_{sat} represents the resistance of the depletion transistor in the saturation region, while R_{lin} corresponds to the resistance of this transistor in the linear region. Neglecting the effect of voltage drops along the column, the control line high voltage can be approximated by:

$$V_{CONTROL} = \frac{VCC + (R_{route} + R_{pullup}) \cdot \left(\frac{n_{HI} \cdot V_{OL}}{R_{sat}} + \frac{n_{LO} \cdot V_{OH}}{R_{lin}} \right)}{1 + (R_{route} + R_{pullup}) \cdot \left(\frac{n_{LO}}{R_{lin}} + \frac{n_{HI}}{R_{sat}} \right)}$$

where n_{HI} is the number of datapath elements with logic inputs high, n_{LO} is the number of datapath

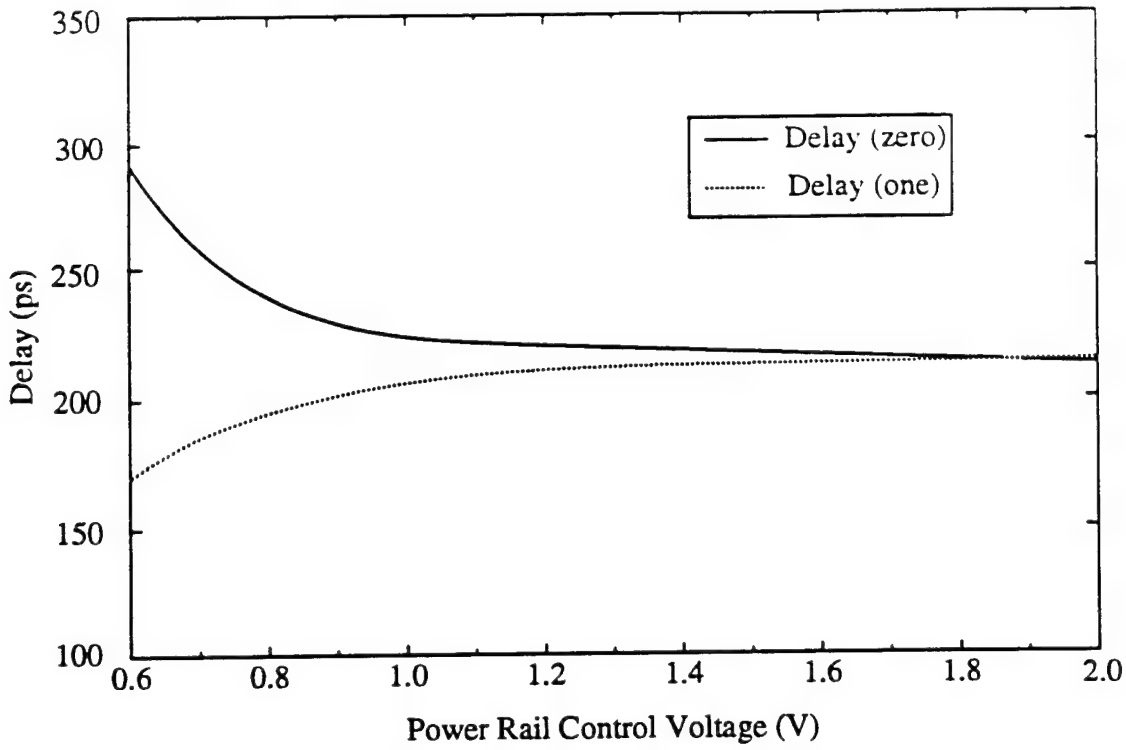


Figure 4.17: Delay for passing a one and a zero through a two-stage power-rail gate as a function of power rail control voltage.

elements with logic low inputs, V_{OL} is the DCFL output low voltage and V_{OH} is a DCFL output high voltage. To maximize the control line high voltage the routing resistance, R_{route}, and the column driver resistance, R_{pullup}, should be minimized while the saturation and linear resistances of the PRL inverter pullup devices should be maximized. Maximizing R_{pullup} will minimize the coupling between the power rail signal and ground. Using minimum-sized input stage depletion pullup transistors will maximize the R_{lin} and R_{sat} terms. The R_{route} and R_{pullup} terms are minimized by using wide power rail buses and wide column driver pullup transistors.

A PRL gate will behave correctly as long as the control line high voltage is above 0.6V. As with most ratioed logic styles, a DCFL gate exhibits a larger delay to raise the output with a depletion load than to lower it through an enhancement transistor. The asymmetrical delays tend to balance after passing successive inverting stages. Fig. 4.17 shows the delay for passing a zero and passing a one through a gate formed by a PRL inverter followed by a DCFL inverter as a function of the control line voltage. Control line voltages below 0.8V cause the gate to show a substantial increase in delay. For control voltages above this, the worst delay is still within 10% of

its value at 2.0V. The lower current drive of the depletion load transistor in the first stage causes the propagation delay for passing a zero to be significantly greater than the delay for passing a one. This asymmetry in propagation delays is reduced in the second stage where the larger of the two delays from the first stage is followed by the smaller of the delays in the second stage. This effect reduces the sensitivity of the circuit's total propagation delay to the power rail high voltage.

The DCFL and PRL topologies are essentially identical. Because the PRL inverters have only one instead of two pull-down transistors they have somewhat smaller parasitic capacitances. So long as the depletion load transistors in the PRL inverters and the control line drivers are sized so that the control line high voltage is at least 0.8V, the two configurations exhibit similar delays from data-input to data-output.

4.5.2.2 Control Signal Input to Data-Output Delays

The simulation results of Fig. 4.13 show that PRL gates achieve smaller control to data-output delays than DCFL; we explain why in this section. In the discussion that follows, we refer to the process of activating a gate as allowing data to pass through it.

The control line of the DCFL datapath module (see Fig. 4.6) is driven by an FFL gate which provides a very large transient current when charging the output line. This leads to a short delay for de-activating the DCFL gate (or forcing the output of the driven gate low). The control line of the PRL datapath is connected directly through a depletion transistor to the output of the driven gate (Figs. 4.4 and 4.7). Discharging the PRL control line simultaneously discharges the associated PRL inverter output, thereby rapidly de-activating the driven gate. The delay required for de-activation in both logic styles is very similar.

To activate a DCFL gate, the control input of the gate must be pulled low. The output of the DCFL gate can conditionally go high, depending upon the data-input. If the gate does switch states, it does so only after the enhancement pull-down FET driven by the control line is turned off. The rate that the output charges is determined by the load current.

The PRL gate, on the other hand, has a much more direct path from the control signal to the gate output. Similar to the DCFL gate, the only transition to consider is the one in which the data-input to the PRL gate is low, causing the output of the PRL inverter to go high. In this case,

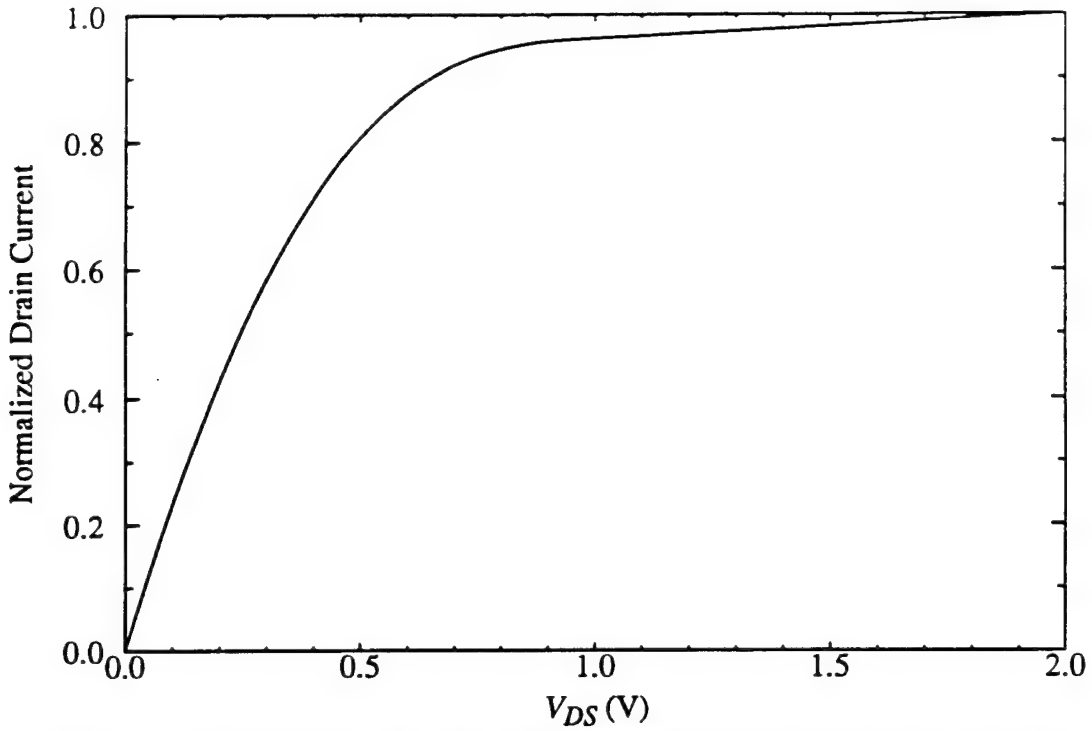


Figure 4.18: Normalized I_D-V_{DS} curve for a gate-source connected depletion transistor.

the depletion load initially has a small voltage drop across it, unlike a depletion load transistor in a DCFL inverter. Fig. 4.18 shows a normalized I_D-V_{DS} curve for a gate-to-source connected depletion transistor with V_T = -0.8V. In DCFL gates, the drain-to-source voltage across a depletion transistor is always between 1.4V and 2.0V, keeping the transistor in saturation. By contrast, as the control line of a PRL gate is raised, V_{DS} of the depletion transistor starts to increase, moving the depletion transistor from the linear region of operation to the saturation region. Fig. 4.19 shows a transient simulation of this gate with different DCFL inputs as the PRL signal is raised and lowered. The data and column driver-inputs are shown in 4.19(a). The power rail control line and the output of the PRL inverter stage are shown in 4.19(b). The transient supply current demanded by the column driver is shown in 4.19(c). The associated drain-to-source resistance of the depletion transistor was calculated by differentiation and plotted in 4.19(d). This figure shows that as the power rail control signal is raised, the depletion transistor initially exhibits a small resistance. This causes a large transient demand for supply current which acts to charge the output node of the PRL inverter quickly. As the drain-to-source voltage of the depletion transistor rises,

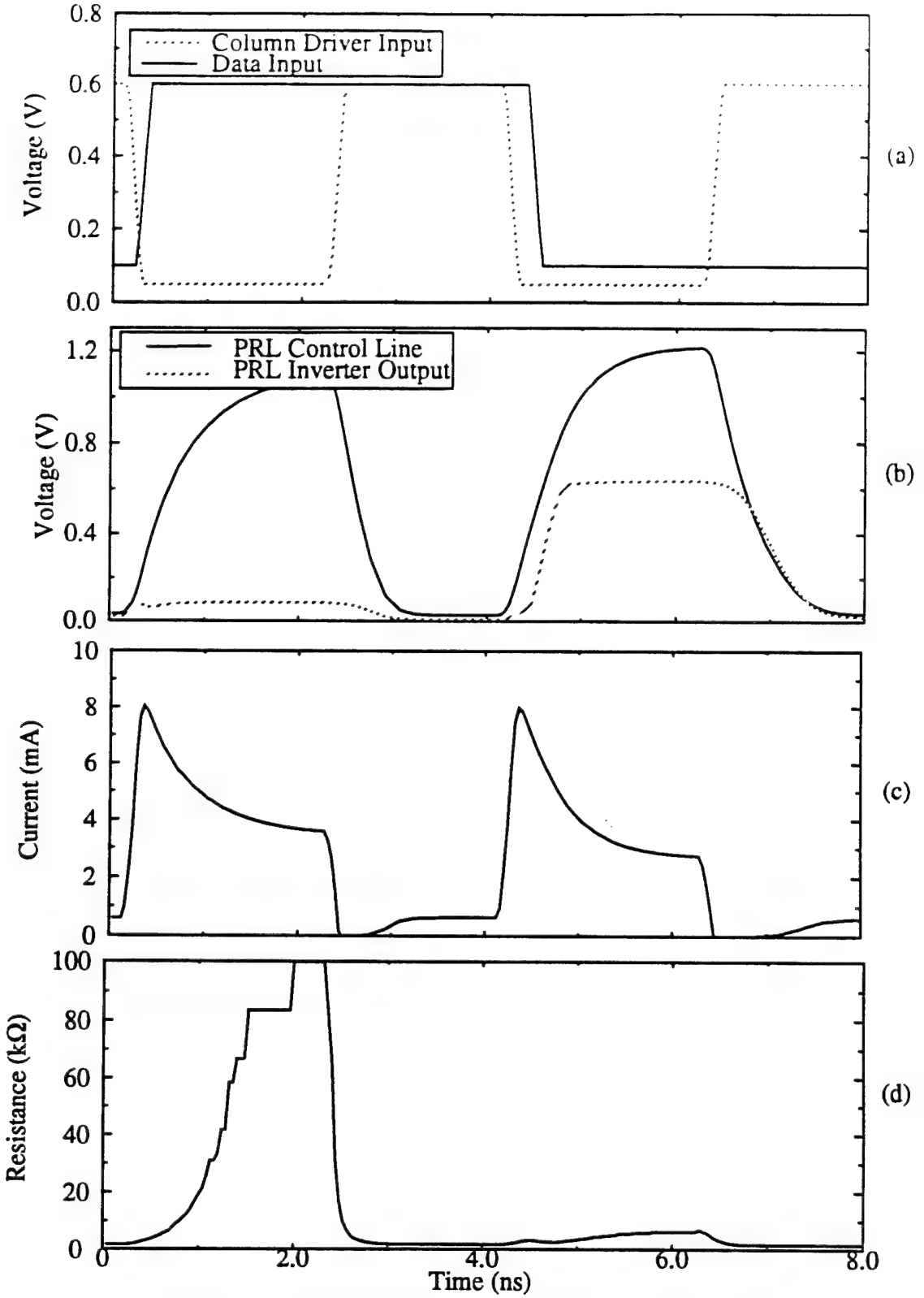


Figure 4.19: Transient simulation of driving a PRL control line. (a) column driver and data inputs; (b) PRL inverter output and PRL control line signal; (c) column driver supply-current demand; (d) PRL inverter depletion transistor resistance.

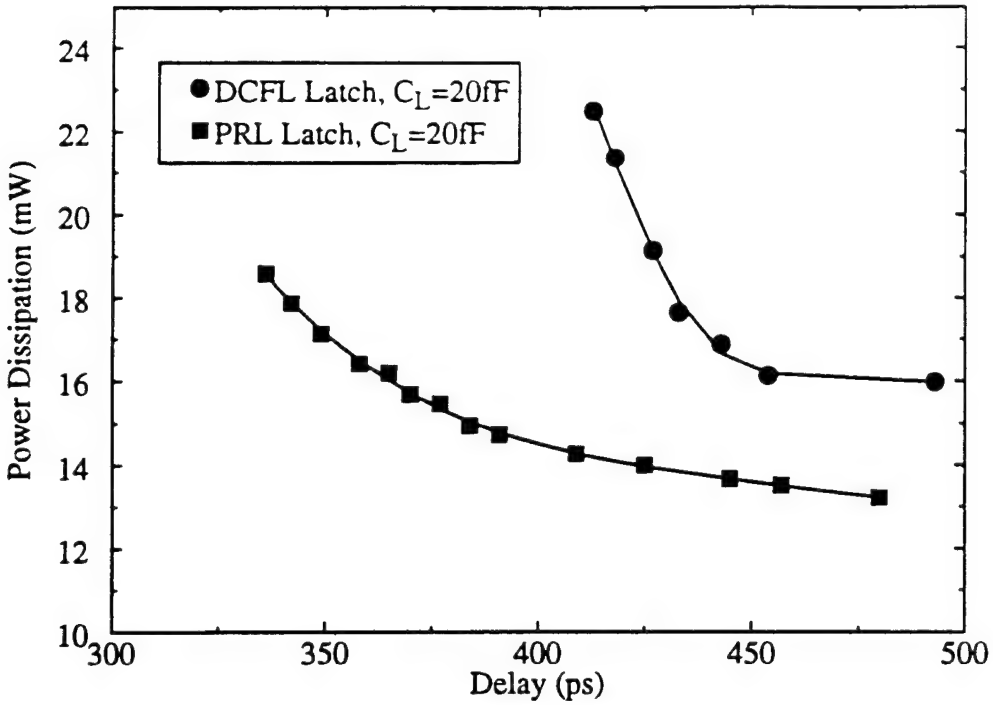


Figure 4.20: Power-delay curves for a lightly loaded DCFL and PRL latch, C_L=20fF.

the gate-source voltage of the column driver pullup transistor starts to decrease, thereby reducing the current supplied.

During the first 4ns of the simulation the datapath element is passing a zero to the output of the input stage. There is no associated delay since the output was already a zero. In the second 4ns the control line is used to drive the output of the first stage high. Since the depletion transistor exhibits a small resistance during this time, the output is rapidly charged as the control line voltage is raised.

4.5.3 Lightly Loaded Latch

The power-delay curves generated for 20fF-loaded latches are shown in Fig. 4.20. The overall power dissipation of both types of latches at this smaller load capacitance is about half of what is required for the 100fF-loaded latches. We would expect power savings much closer to the upper bound for the lightly loaded case. At the smallest propagation delay point shown for the DCFL latch, it dissipated 22.5mW as compared to 14.5mW for the PRL latch, for a savings of 35%. Again we see that the savings are larger than the "upper-bound" simply because of the smaller delays achieved by the PRL circuitry.

Table 4.2: 32-bit Datapath latch power components.

State of Latch	PRL Latch	DCFL Latch
Transparent	25.6mW	38.7mW
Latched, Transient	20.8mW	43.4mW
Latched, Quiescent	20.2mW	42.8mW
Average	**23.3mW**	**41.1mW**

4.5.4 Infrequently Transparent Latches

Power rail latches can be used to provide even larger power savings in situations where data is held in the latch for comparatively long periods of time. For example, queues and buffers, which use latches, tend to use individual registers infrequently, and most registers are usually in the latched, rather than transparent, state. The area overhead for implementing these functions with SRAMs is too great, so they are normally implemented as datapath latches. The average power dissipation, P_{avg}, is a function of the frequency of latching and is given by

$$P_{avg} = \frac{P_{transparent} + P_{latched-transient} + (n-1) \cdot P_{latched-quiescent}}{n+1}$$

where $P_{transparent}$ is the power dissipated when the latch is transparent (i.e., inverters turned on), $P_{latched\text{-}transient}$ is the power dissipated in the first clock phase when the outputs are latched (input inverters turned off), $P_{latched\text{-}quiescent}$ is the power dissipated in the latch during a clock cycle when the latch is just holding previous data, and n is the number of clock phases in which the latch is kept exclusively in the latched state.

Table 4.2 is a breakdown of power dissipation for a 32-bit datapath DCFL and PRL latch that both achieved the same delay from clock input to data-output while driving a 100fF load. For this design point, and assuming the latch spends equal time in each state, the PRL datapath achieved a 43% power savings. The quiescent latched power dissipated by the PRL latch is only 20.2mW compared to 25.6mW of power when the latch is transparent. The average power dissipated by a latch that has a 50% clock duty cycle was simulated as 23.3mW. Thus, the lower bound for the power dissipation that an infrequently transparent latch can achieve is 20.2mW, which is 87% of the average power dissipated in a latch that is always switching. By contrast, the

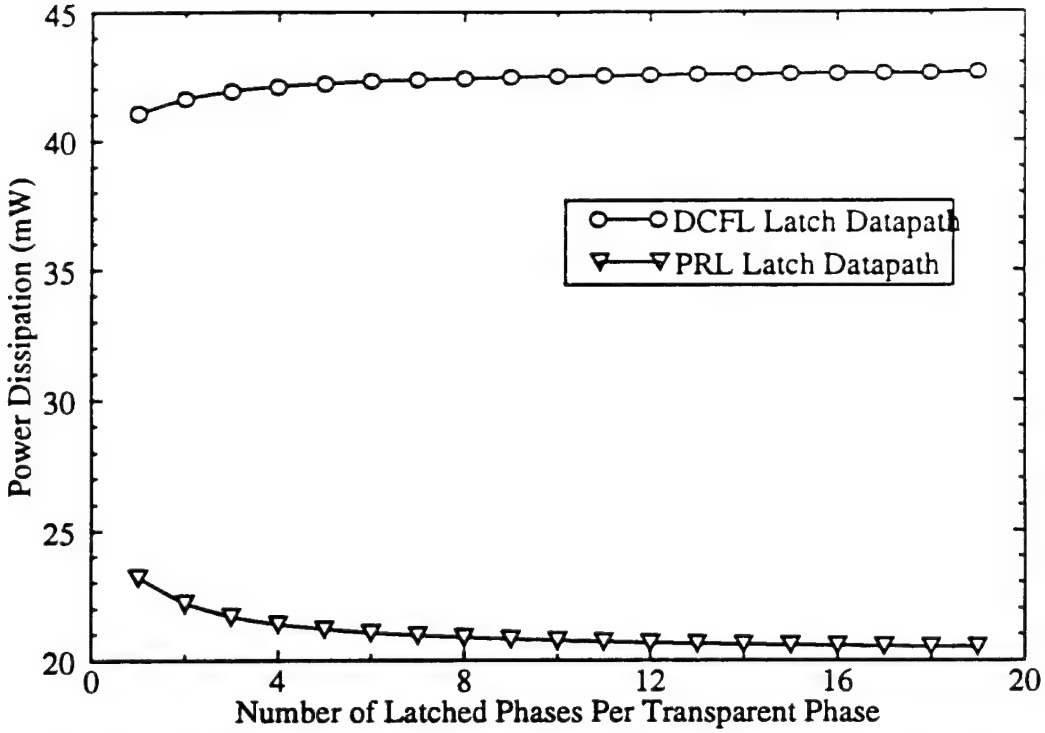


Figure 4.21: Power dissipation of an infrequently transparent DCFL and PRL latch.

power dissipation of an infrequently transparent DCFL latch actually increases, as seen in Fig. 4.21. This is because the clock line of a DCFL latch supplies current to the clock inputs in the latched mode.

This concept of logically shutting down circuitry could have a much farther-reaching impact on processors in which entire pipeline sections, including latches, muxes, flip-flops, etc. are often not used because of stalls. In a multi-issue machine, a lack of parallelism in the instruction stream often causes entire pipeline stages to sit idle. In these cases, *idle* control signals could be used to turn off all active control signals associated with PRL datapaths, thereby achieving significant power savings. An interesting architectural investigation would be to determine the power savings that may result from such a scheme.

4.6 A PRL Flip-Flop

Another example of a circuit which can exhibit significant power savings is a master-slave flip-flop, shown in Fig. 4.22. In this circuit, at any given time either the master section or the slave section will benefit from having one half of its gates turned off. The master section output gates

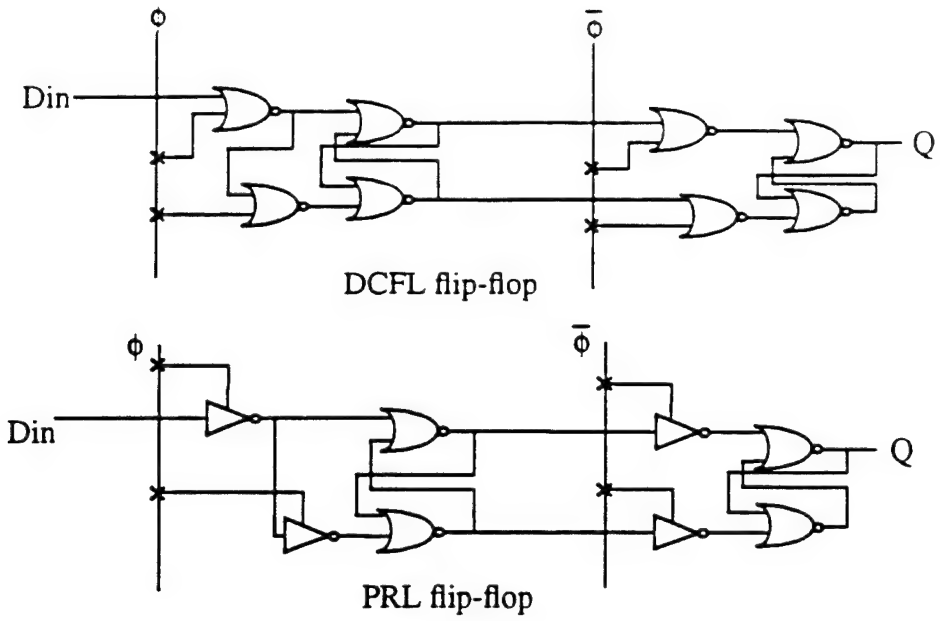


Figure 4.22: DCFL and PRL flip-flops.

can be kept minimum-sized since they only have to drive a small capacitance local to the cell. The only high-drive gate is thus the output NOR gate. Since the master latch is lightly loaded and the output of the slave latch is moderately loaded, we expect that the net power savings would be close to the average of what was found for the lightly loaded and heavily loaded latches studied above.

A comparison with the DCFL master-slave flip-flop is given in Fig. 4.23. This comparison was performed with a 32-bit datapath assuming each of the elements to be loaded with 100fF. The power dissipation measured was the average value over latching a zero and a one at 250MHz using a 2.0V supply. Two delays were used for comparison between the implementations. The first is from the input data to the output of the master section of the flip-flop, representing a data-setup time. The second delay is measured from the latching edge of the driver of clock signal $\overline{\phi}$ to the output Q. In the circuit of Fig. 4.23, the optimization was done on clock-to-Q delays. The data-to-master Q curves represent the delays from the flip-flop data-input to the master stage latch output for those designs that achieved the minimum clock-to-Q power-delay products.

The PRL implementation exhibits a larger delay from the data-input to master latch output than the DCFL implementation (by about 40ps). This is due to a reduced power rail voltage reaching the gates, as described in section 4.5.2. The clock-to-Q delays are smaller (by as much as

90ps) for the PRL flip-flop. The reasons for this smaller delay were also described in section 4.5.2. More importantly, the PRL flip-flop can achieve very comparable performance to the DCFL flip-flop at about half of the power dissipation. In contrast to the latch described in the previous section, the flip-flop has one clock signal or the other active all the time. In the DCFL circuit, this means that an FFL driver is always on, demanding a large transient current at each rising clock edge and driving a large diode load. The column driver component of the total power dissipation causes the DCFL datapath flip-flop to consume much more power than the PRL flip-flop.

4.7 A PRL Mux-Latch-Buffer

In a VLSI design, a mux or latch followed immediately by a buffer to provide high drive is much more common than a mux or a latch in isolation. A common combination of cells found in the Aurora II processor is a two-input mux followed by a latch and a buffer. This logic, integrated into the mux-latch-buffer (MLB) cell shown in Fig. 4.24(a), is often used to give latches a scan capability. A power rail logic implementation of this cell is shown in Fig. 4.24(b). In addition to using power rail logic for the multiplexor select and clock lines, this implementation also uses diode D1 to provide both DCFL and super-buffer logic levels, an idea novel to this thesis.

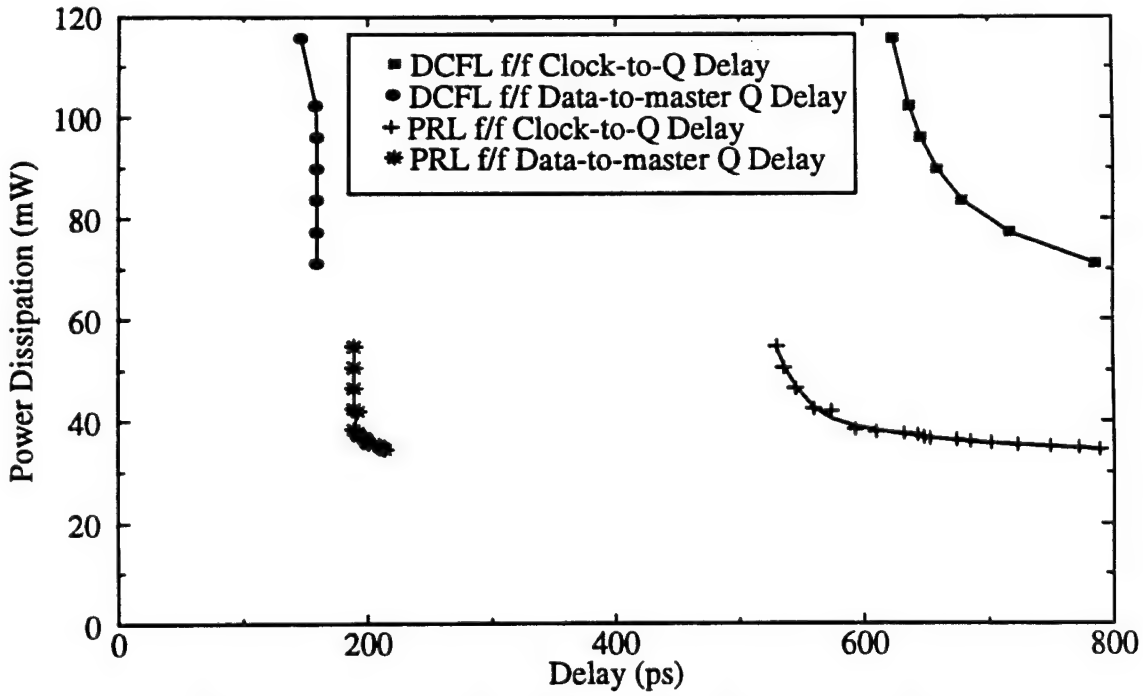


Figure 4.23: Characteristic DCFL and PRL flip-flop power-delay curves.

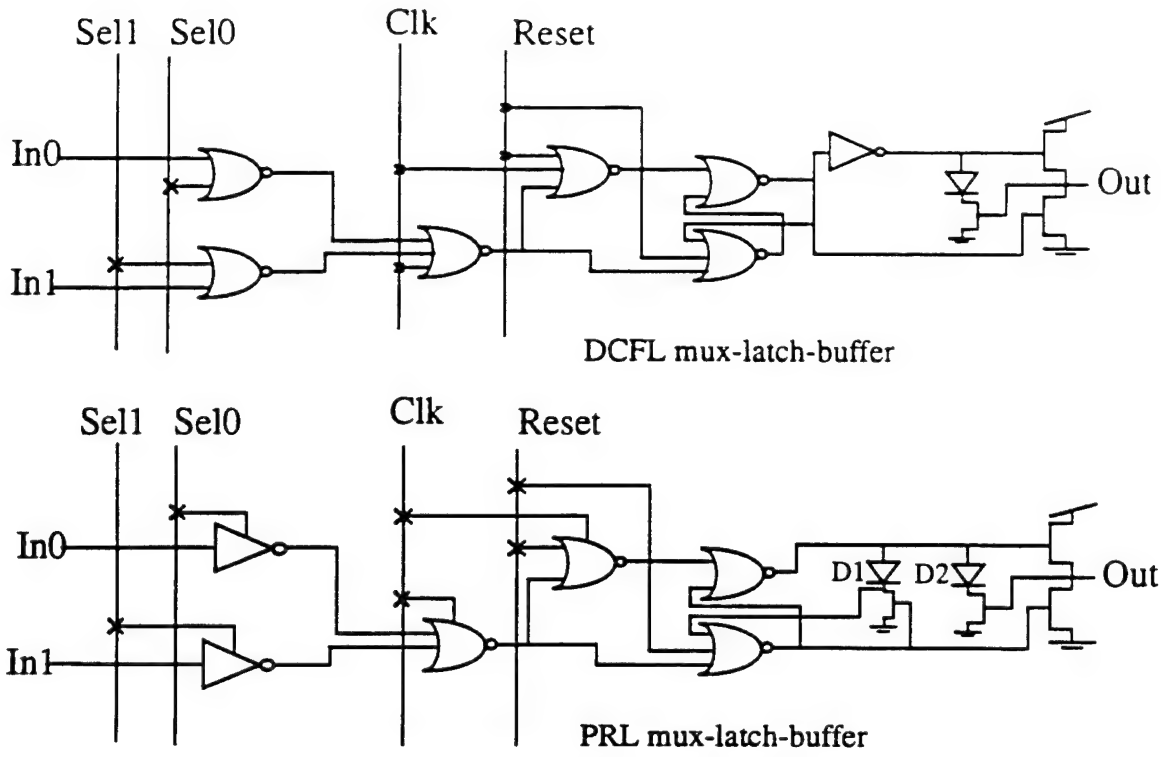


Figure 4.24: A DCFL and a PRL mux-latch-buffer.

In moderately loaded PRL gates, the output stages had to be sized up to drive inter-cell capacitance, leading to smaller overall savings than one would expect from power rail logic. In the PRL mux-latch-buffer, inverters, NOR gates, and latching NOR gates drive only small capacitances local to the cell, and hence the potential power savings are greater. This advantage is offset in the example presented here by additional delay incurred through using smaller column drivers in order to achieve a compact layout, leading to lower power rail control line high voltages.

The mux-latch-buffer is characterized by three sets of delays. These are the clock-to-output delay, data-in to data-output delay and select-to-output delay, where the delays from clock and select are measured from the inputs to the respective column drivers. For inter-latch critical paths of more than 1.2ns, the clock-to-output delay is less important than the select-to-output and data in-to-output delays.

Fig. 4.25 shows the power-delay graphs for a 32-bit datapath mux-latch-buffer circuit. The power dissipation used for comparison is the average of the power dissipated when the clocks are running and the output data is toggling. Both the PRL and DCFL MLB output feedback-FET-logic buffers were loaded with 3pF of capacitance and 20μm of diode load. The FFL buffer

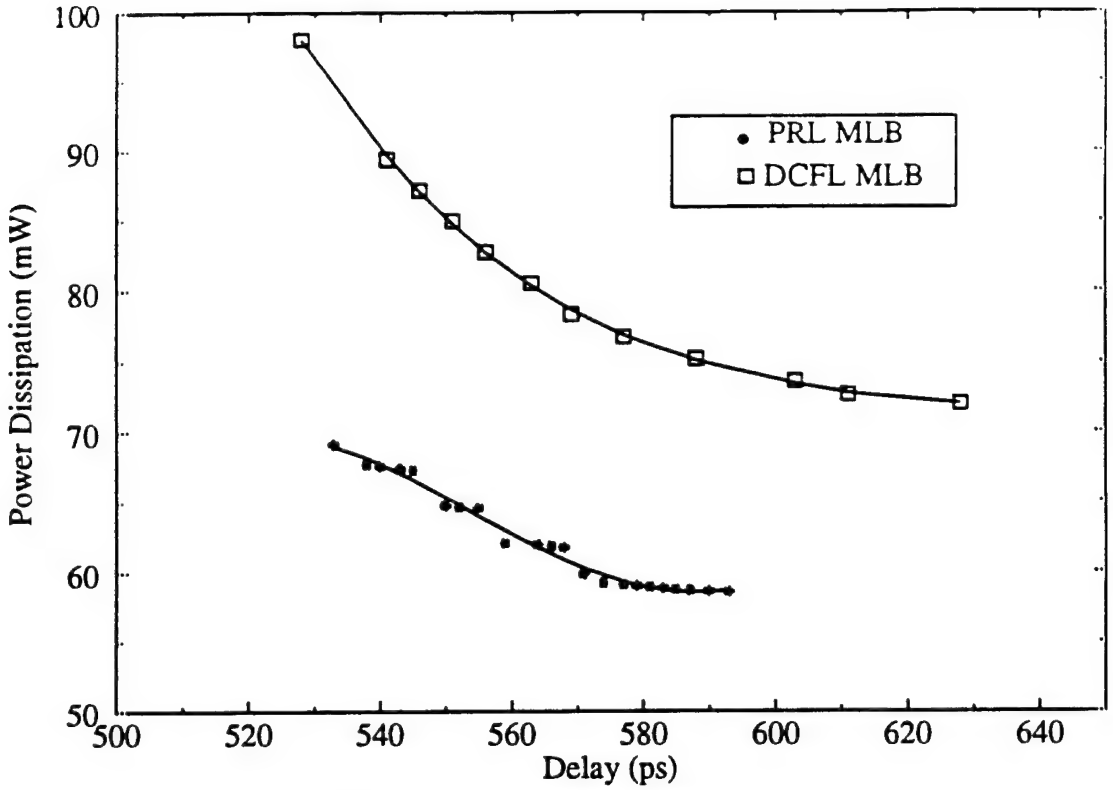


Figure 4.25: DCFL and PRL mux-latch buffer power-delay curves.

exhibits a large transient demand for power when the output data is switching from a zero to a one. On average this transition occurs once out of every four possible data transitions (including one-to-zero, zero-to-one, one-to-one and zero-to-zero), which were considered equally probable in this comparison. The comparison was made while also switching the select line once per clock cycle.

Previous sections have shown that the control line-to-output delay is smaller for the PRL circuits than for DCFL circuits. This is also the case for the PRL mux-latch-buffer circuit. In this section, the basis of comparison was the data-input to data-output delay, assuming a transparent latch and a stable select signal.

Due to the dependence of the data-input to data-output delay on the power rail control voltage level (as demonstrated by Fig. 4.17), PRL can have somewhat longer delays than DCFL. Due in large part to the elimination of one logic stage with the introduction of the level shifting diode, the PRL gate achieves data-in to data-out delays comparable to the DCFL circuit. Despite the fact that the PRL circuit also has a high-drive buffer at the output, it achieves significant power savings for the same delays as illustrated by the power-delay curves of Fig. 4.25. While the curves

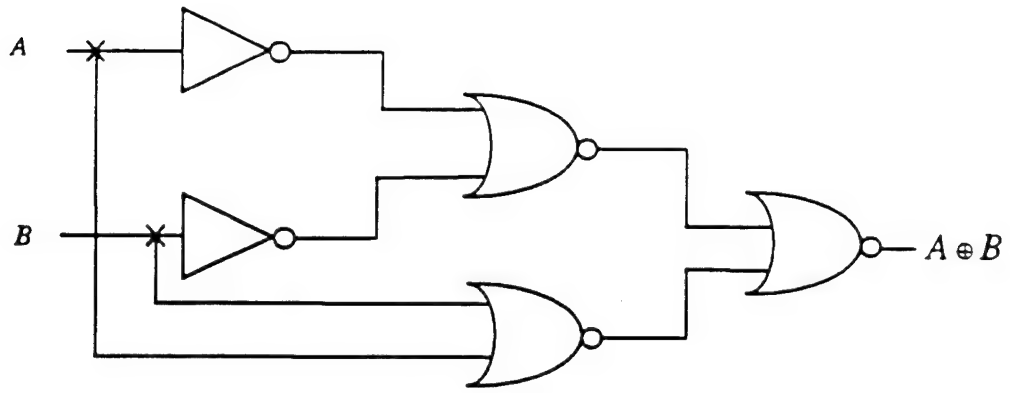


Figure 4.26: A DCFL XOR gate.

do not follow iso-power-delay product curves, the smallest power-delay product achieved for the DCFL MLB is 44 pJ, whereas, for the same delay, the PRL implementation achieves a power-delay product of 34 pJ, resulting in a 23% reduction in power-delay product.

This circuit example demonstrates that even when integrated with high-drive output buffers, power-rail logic datapaths can offer significant power savings over their DCFL counterparts.

4.8 A PRL Exclusive OR Gate

In the previous sections, power rail logic datapath circuits have been presented which were all derived from DCFL circuits by replacing two-input and three-input NOR gates with PRL inverters and PRL NOR gates. PRL circuits can also be used to realize complex logic functions that do not require column drivers to block or pass signals. In this section we present an exclusive-OR gate in which DCFL inverters are used to drive the power rails of logic gates. This technique can be applied equally well to other complex gates. As with many circuit design techniques, the creativity of a circuit designer will influence how effectively this style of design is used.

A compact DCFL exclusive-OR gate is shown in Fig. 4.26. This gate has a delay of three levels of logic and requires thirteen transistors; at any given time all five gates are drawing current. The PRL XOR gate is presented in Fig. 4.27. This implementation requires eleven transistors and two diodes. In this circuit, the outputs of DCFL inverters 1 and 2 are used to drive the power rail inputs of inverters 3 and 4. The diodes are sized such that both DCFL and PRL logic levels for $\overline{A}$ and $\overline{B}$ can be derived from the same wire. The outputs of inverters 3 and 4 are used to drive the

buffered FET logic (BFL) output stage.

One advantage of the PRL XOR gate is that it is integrated with a high-drive BFL gate, allowing it to efficiently drive large capacitances. Only the two input inverters are always on. In the BFL output stage, at most only one of the pullup transistors is on at a time. If the XOR result is zero, then neither of the pullups will be on. Thus the circuit draws a current that is data-dependent and there will be either two or three gates drawing power at any given time compared to five for the DCFL XOR gate.

The disadvantage of the PRL XOR gate is that it is not tolerant of large supply voltage drops. The depletion transistor on an input inverter must drive the following PRL inverter pullup gate as well as the BFL OR-gate. The high voltage requirement of the BFL OR-gate prevents this circuit from working at supply voltages below 1.3V.

The methodology of section 4.2 was applied to the DCFL and PRL XOR gates for comparison. The results of this comparison are shown in Fig. 4.28. As the figure shows, the PRL XOR gate exhibits better power-delay products than the DCFL XOR gate with a 2V power supply. The sensitivity of the output high voltage and delay through the PRL exclusive-OR gate is shown in Fig. 4.29. Below a supply voltage of 1.3V, the output high voltage noise margin drops low enough that across process and temperature it may not be able to properly drive the subsequent stage. Because the slower of the two propagation delays is the output-high to output-low transition, the delay of the XOR gate actually decreases with decreasing supply voltage down to 1.6V. Below 1.6V the output-low to output-high delay starts to dominate. At a supply voltage of 1.3V the PRL delay is only 20% larger than its value at 2.0V.

Rather than using a BFL output stage, PRL inverters 3 and 4 can be used to directly feed a NOR gate, thus realizing the exclusive-NOR function. The resulting gate has much better supply

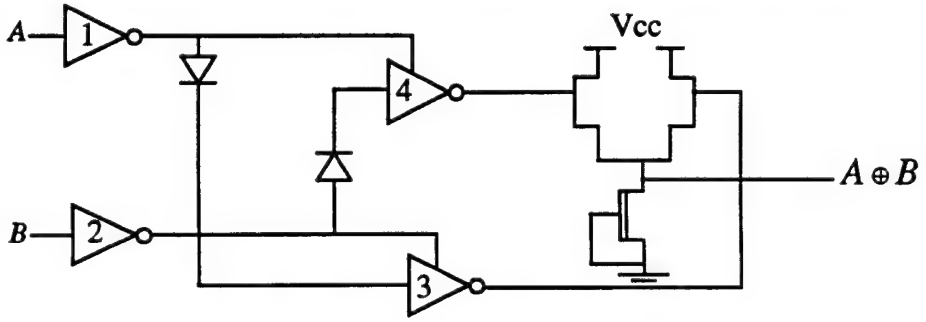


Figure 4.27: A PRL XOR-gate.

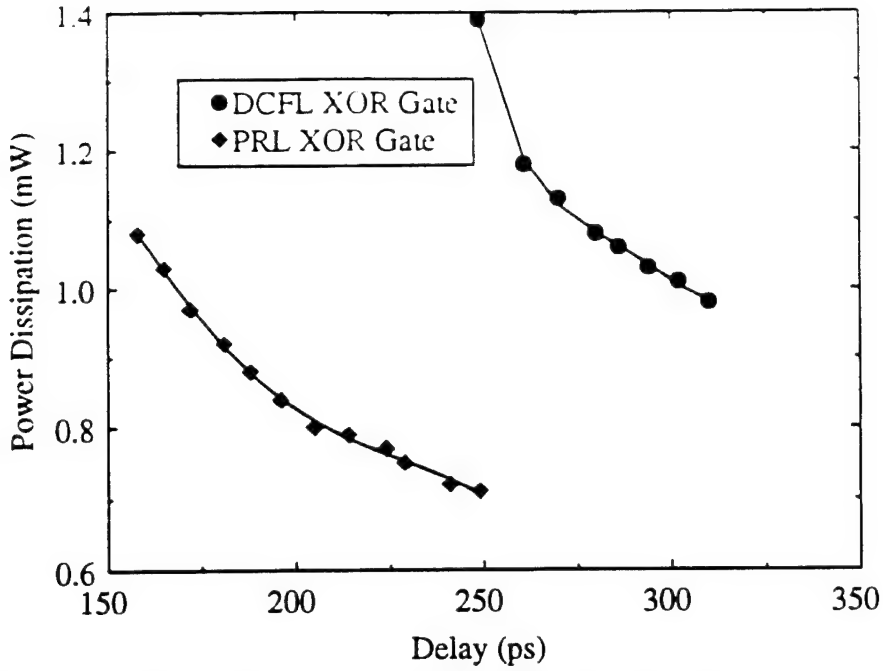


Figure 4.28: Power-delay curves for a DCFL and PRL XOR gates with a 2V supply.

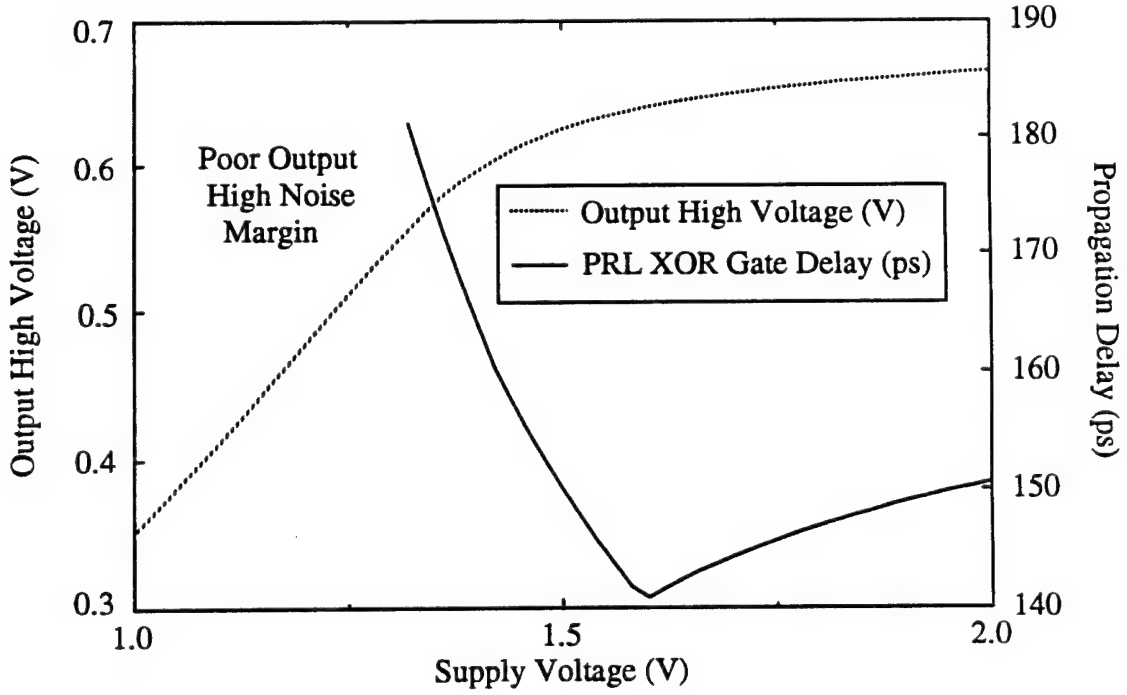


Figure 4.29: Sensitivity of the output high voltage and propagation delay of a PRL XOR gate to supply voltage.

voltage tolerance than the XOR gate. The XOR gate example of this section can find applications in adder circuits and comparators, which are used extensively in all types of digital circuits.

Table 4.3: Typical measured results for the 32-bit DCL and PRL barrel shifters.

	DCFL Barrel Shifter	PRL Barrel Shifter
Area	1.047 mm^2	0.931 mm^2
Power Dissipation	217 mW	165 mW
Delay per cycle	1.22 ns	1.06 ns

4.9 Demonstration Vehicle

A test chip was designed, fabricated, and tested to demonstrate PRL circuitry. A die photo is shown in Fig. 4.30. The chip consists of one 32-bit barrel shifter designed using DCFL and one designed using PRL. The barrel shifters were constructed to allow shifting of a 32-bit word by 0 to 31 bits using a cascade of three multiplexors. A schematic of the shifter and a floorplan of the test chip are given in Fig. 4.31. The input 2-to-1 multiplexor is used to load the shifter with either an external barrel-in input bit combined with the lower-order 31 bits of the output or the previously shifted result stored in the output latch-buffer.

Both barrel shifters have been tested and have demonstrated identical functionality. Typical test results measured from a lot of 20 packaged die are given in Table 4.3. One test of the barrel shifter was to apply a pulse to the input of the barrel shifter, set the shift-amount to 1, open both latches and wait for the signal to propagate through 32 cycles of the shifter until it finally exited through the barrel-out signal. This test gives a good comparison of the delays of the two circuits. Fig. 4.32 is an oscilloscope trace showing a barrel-in signal and the barrel-out signals generated from both the DCFL and PRL circuits on one chip. The power supply of each circuit was made accessible using bare bonding pads to get an accurate comparison of the power dissipations of the two circuits. The PRL circuit, which is approximately 12% smaller, was found to operate 13% faster than the DCFL circuit while consuming an average of 24% less power, resulting in a 34% smaller power-delay product.

4.10 Conclusions

This chapter has presented a methodology for characterizing and comparing circuits that

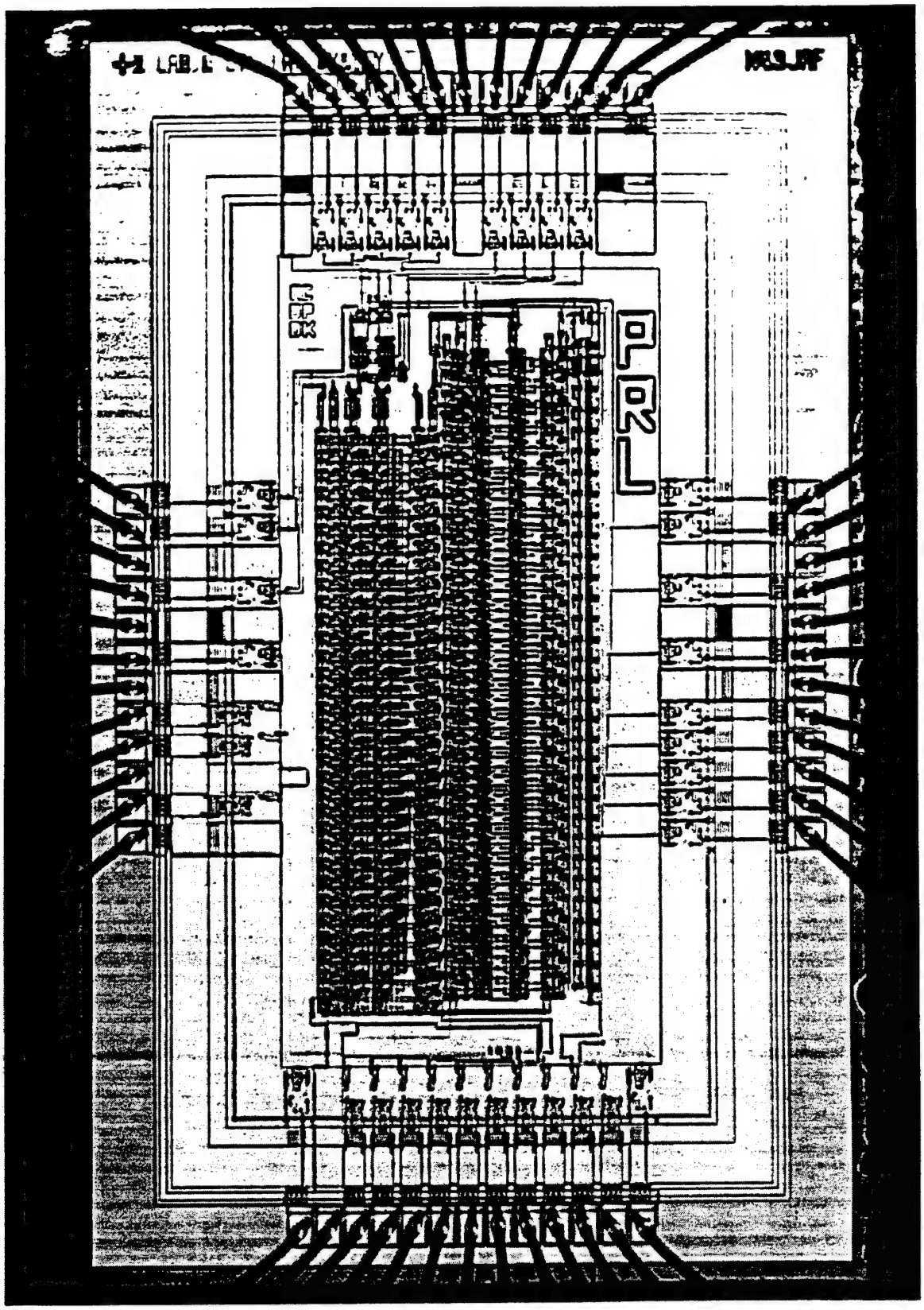


Figure 4.30: Die photograph of the PRL test chip.

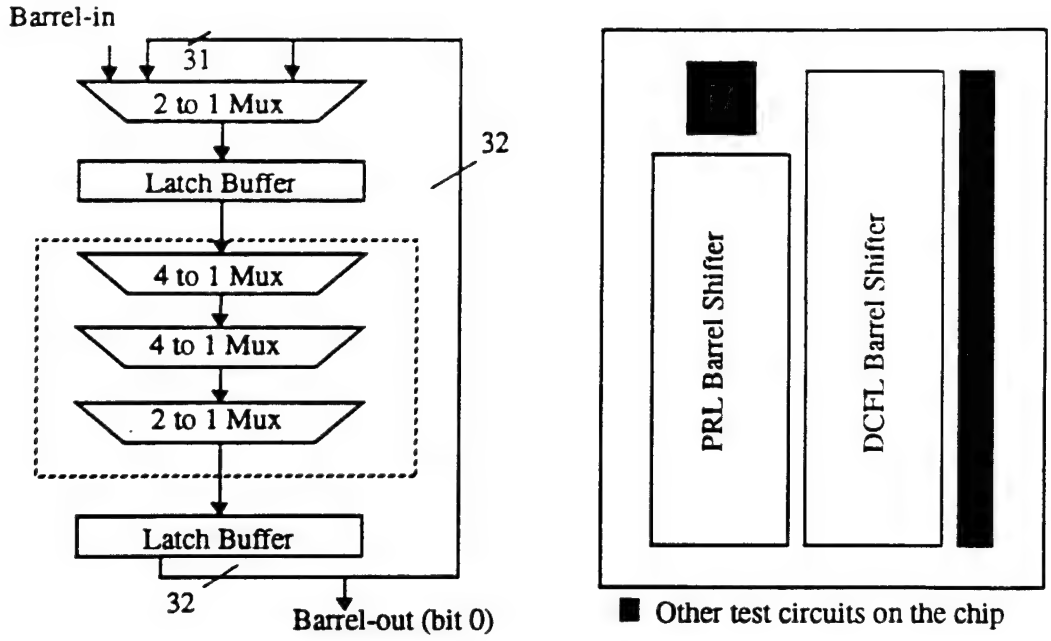


Figure 4.31: Barrel shifter schematic and PRL test chip floorplan.

uses process tolerance as a central part of the analysis. The small logic swings, low noise margins and sensitivity to process variations make such an approach essential for the comparison of GaAs circuits. The methodology was used to evaluate power rail logic which is a new logic style that is novel to this thesis. This logic style offers circuits with a smaller area and up to 40% lower power-delay products than can be achieved with DCFL. The effectiveness of this logic style has been demonstrated for some of the most common digital logic circuit datapath elements, including multiplexors, latches and flip-flops. This approach can be extended to great advantage in pipelined and multi-issue processor systems in which entire datapath sections are either stalled or are otherwise often idle. Power rail logic can also be applied to simplify complex random logic gates, as was illustrated by the exclusive-OR gate.

A test chip containing 32-bit DCFL and PRL barrel shifters was designed, fabricated and tested. The PRL circuit, which is about 12% smaller, was found to operate 13% faster than the DCFL circuit while consuming an average of 24% less power, resulting in a 34% smaller power-delay product. This demonstration vehicle proves the viability of power rail logic.

With the circuit characterization methodology and programs in place, we now have a building block from which process-tolerant automatic transistor sizing algorithms can be

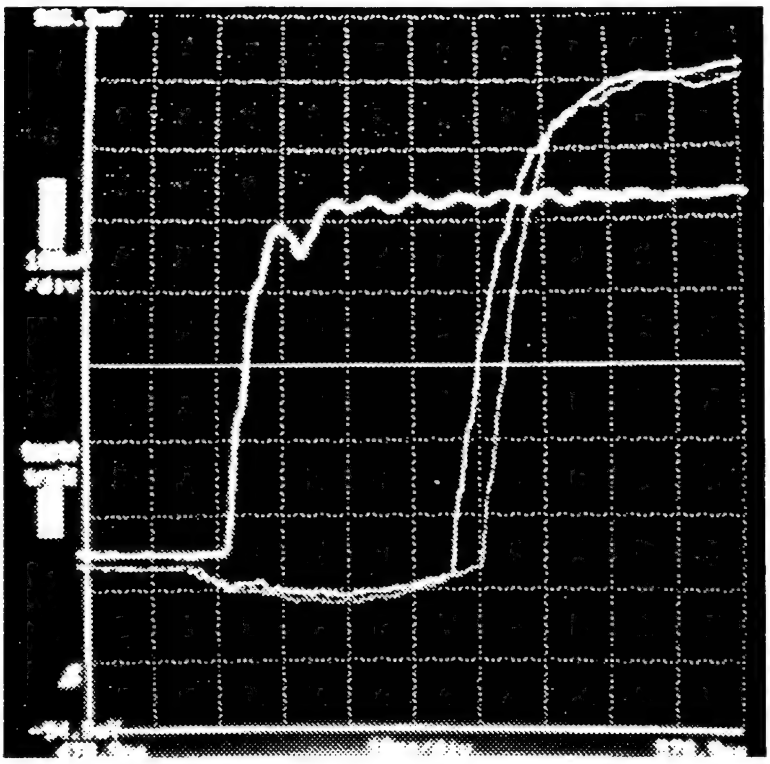


Figure 4.32: Propagation delays of 32 cycles of the PRL and DCFL barrel shifter. (left to right) barrel-in, PRL barrel-out, DCFL barrel-out.Time base: 10ns/div., Vertical: 100mV/div.

developed for use in an SRAM compiler, which is presented in the next chapter.

CHAPTER V
THE AURORA RAM COMPILER

As the integration levels in any technology increase, system designers can afford to put larger amounts of memory on chip. The exponentially large number of possible configurations makes it impossible to build and maintain a complete library of standard RAM configurations. The solution has been the development of software, called RAM compilers, which automatically generate layout for a RAM of a specified size and organization.

The preceding chapters have described an evolution of circuits and design methodologies required to produce higher speed, lower power and process tolerant circuits in E/D MESFET GaAs. The process tolerant design methodologies have a much broader scope and can be applied to circuit design in any processing technology. In this chapter, the Aurora Ram Compiler (ARC), which builds on the circuit innovations and process tolerant design methodologies is presented.

RAM compilers have grown in sophistication from early implementations which were merely layout generators[Swa86] to more recent implementations which perform automatic buffer sizing to meet delay requirements[Shi91]. In Section 5.1, the flexibility, design methodologies, delay calculation approaches, and general goals of commercially available SRAM compilers are described. This background helps to differentiate the ARC from these previous efforts.

The major components used in the ARC are described in Section 5.2. The impact of circuit parameters on the speed, power, and noise margins of the memory are described in this section. ARC consists of a physical design module and a transistor sizing module. In Section 5.3, the structure of the compiler and the interactions between these two modules are described. The physical module, developed by David Kibler and Mark Roberts, is described in Section 5.4.

The transistor sizing module, described in Section 5.5, determines transistor sizes that will

produce an SRAM that is tolerant of process variations and meets target read and write times. In Section 5.6, several constraints that are necessary to meet these objectives are presented. These constraints can only be checked with a simulator that accurately models effects that degrade circuit performance and result in low noise margins. Section 5.7 describes such a model. In Section 5.8, the algorithm used to explore the SRAM transistor size space is presented. This is followed, in Section 5.9, by examples of memories that have been generated with the compiler.

In addition to being a valuable design tool, ARC is also a useful research tool. The compiler can be used to accurately predict the impact of processing parameters such as sheet resistances, line capacitances, process control and process enhancements on SRAM area, speed and power; this is possible because the delay calculations and power estimations are not tied to a particular process. Thus, this tool can be used to accurately predict which trends in technology will have the most significant impact on memory performance. Results of this analysis are presented in Section 5.10.

5.1 Introduction

The first reported RAM compiler was developed by Texas Instruments in 1986[Swa86]. The compiler, called RAMGEN, was primarily a layout generator for RAM, ROM, and PLAs. In addition to generating the layout, RAMGEN also generated simulation models, symbols and datasheets. The datasheet information included setup and hold times, read-access and write times, expected supply current, and minimum supply voltage. These values were determined by curve fitting for a particular RAM size, with parameters obtained from fabricated memories.

Since this first effort, RAM compilers have become more sophisticated. The Memorist compiler, developed by Motorola and Mentor [Tou92], and the multi-port high-speed RAM compilers developed by Cascade Design Automation [Shin90], are CMOS SRAM compilers which offer more advanced features, allowing greater flexibility to the user.

The main goals of the Memorist compiler were to achieve high performance memories and to provide flexibility to the user of the compiler. The high-performance was achieved with extensive use of dynamic logic. CMOS allows a multitude of physical organizations, defined by the number of rows, columns, and column-folding, that can all achieve the same logical organiza-

tion. The Memorist compiler takes advantage of this to provide flexibility to the user by presenting the user with a wide range of RAMs which achieve different speed and power dissipations based on different physical organizations. The users can then select the RAM with the combination of speed, power, and area that best suits their needs. Delay and power estimations can be performed with this compiler by using lookup tables and interpolation for rapid estimation, or by simulating an extracted SPICE netlist for a more accurate evaluation.

The Cascade Design Automation (CDA) RAM compiler was built with a substantially different design methodology to achieve a different set of goals. The primary objective was to build a compiler in a generic framework that could generate high performance memories in several manufacturer's processes. The high performance goal was achieved by using dynamic logic extensively and by providing automatic buffer sizing using the following algorithm. First, all buffer are initialized to their minimum sizes. Based upon the capacitive loading of the various interacting cells, new buffer sizes are determined to achieve equal rise and fall times. The new capacitances introduced by the larger buffers are then used in a second pass to refine the buffer sizing and delay calculations. The CDA compiler relies on interpolation lookup tables for delay calculations.

The CDA compiler achieves process independence by using the Compiler Development System (CDS) [CDA91]. CDS is a high-level layout description language written in C, in which foundry-specific design rules are described by global variables that are initialized at run time. These variables are used to calculate constraints between polygons such that the designed cells are design-rule-check correct-by-construction, independent of the actual physical design rules.

Both the Memorist and the CDA compilers strive to achieve high performance with extensive use of dynamic logic. In the Memorist compiler, buffers are chosen from a library of buffer sizes based upon the size of the array being driven, whereas n the CDA compiler, the automatic buffer sizing is driven by timing constraints. Both compilers strive to provide flexibility for the user. The Memorist compiler provides this by allowing a choice of table look-up or SPICE delay calculations, and by allowing the user to choose from a wide range of physical organizations that best meet power-delay-area requirements. The CDA compiler uses timing driven buffer sizing and process independence to provide flexibility.

The ARC project shares some of the general goals of these previous compilers and intro-

duces a few of its own. These goals include:

1. Design a CAD tool that encapsulates the knowledge base of a RAM designer into a computer program.

2. Provide an intelligent tool which incorporates all the lessons learned from previous chip designs to ensure circuits with a high yield.

3. Develop a CAD tool which can provide sub-2.5ns memory designs of up to 8kb in size in GaAs E/D MESFETs to satisfy the small-memory requirements of the Aurora system.

4. Build a compiler framework that can easily adapt to changes in processing technology.

5. Develop a framework that allows the systematic trading of power for speed in an intelligent manner to provide timing-driven automatic transistor sizing.

6. Develop a tool which could be used to realistically predict the impact of E/D MESFET processing technology on SRAM speed, power and area.

The ARC project has incorporated as many of the desirable features of existing RAM compilers as possible to help meet these goals. For instance, CDS is used for generating process-independent layout generators. Due in part to the low noise margins in E/D MESFETs and also in part to the goal of developing a flexible framework which could meet goals 3 through 6, the ARC design methodology represents a departure from the methodologies that have been used in previous compilers.

5.2 Circuit Design

The SRAM compiler has read and write ports that are accessed synchronously during each clock cycle. An active low memory clock (MEMCLK) signal is used to initiate the read or write operations at the beginning of each cycle, and then to generate an internal equalization pulse used to precharge the bit lines.

The write operation is specified by the WRITE signal, which is active high. This signal must be supplied at the beginning of the read or write cycle. Based on the state of this signal, either a read or a write operation occurs. Timing diagrams for the read and write cycles are given in Fig. 5.1.

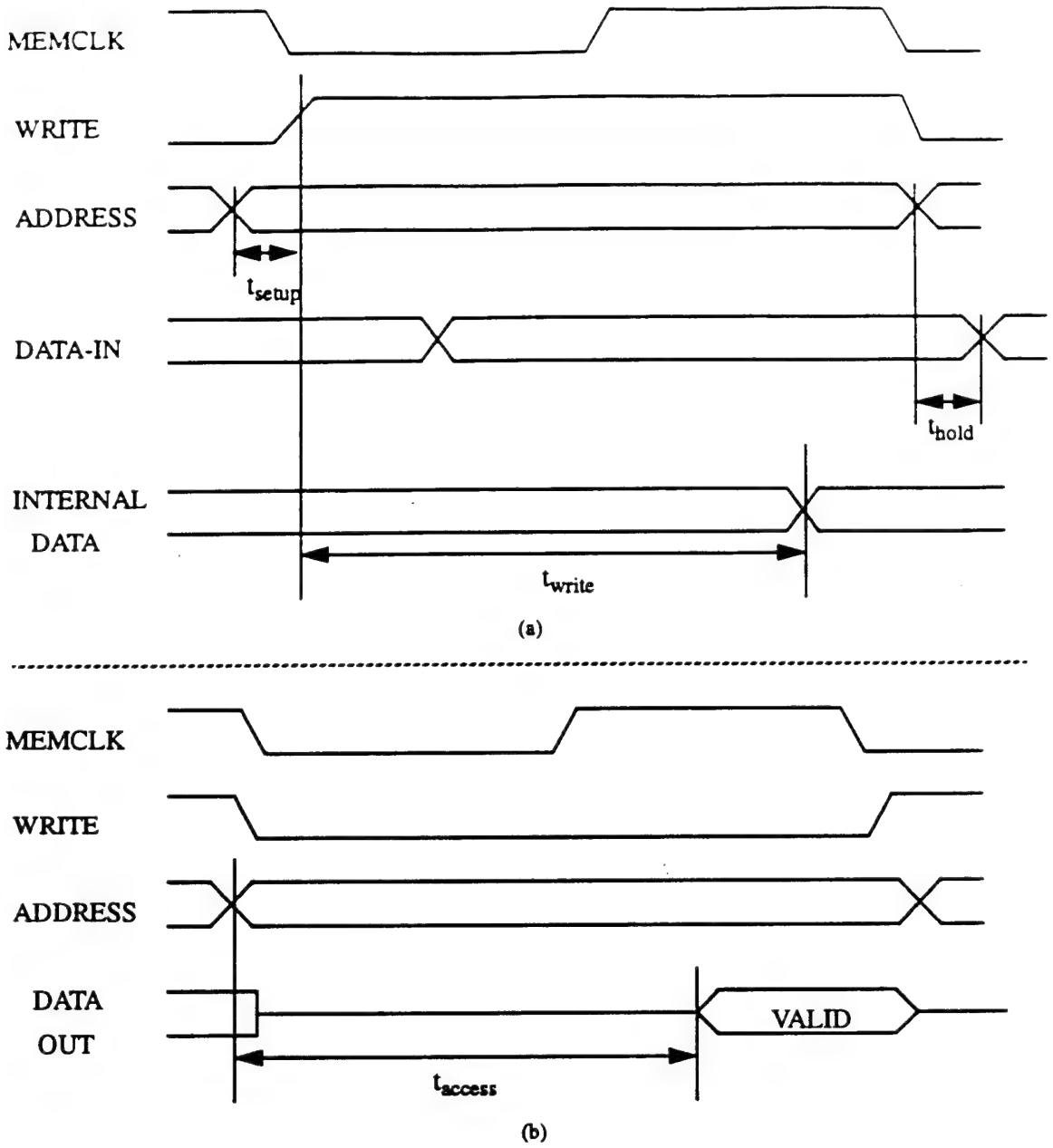


Fig. 5.1: Read and write cycle timing diagrams for the compiler-generated SRAM. (a) write cycle timing diagram. (b) read cycle timing diagram.

5.2.1 Components of the RAM

The major cells used in the RAM are shown in Fig. 5.2. The pulse generator, sense-amplifiers, write circuitry, equalization circuitry, memory cell array, cell ground drivers, and word line

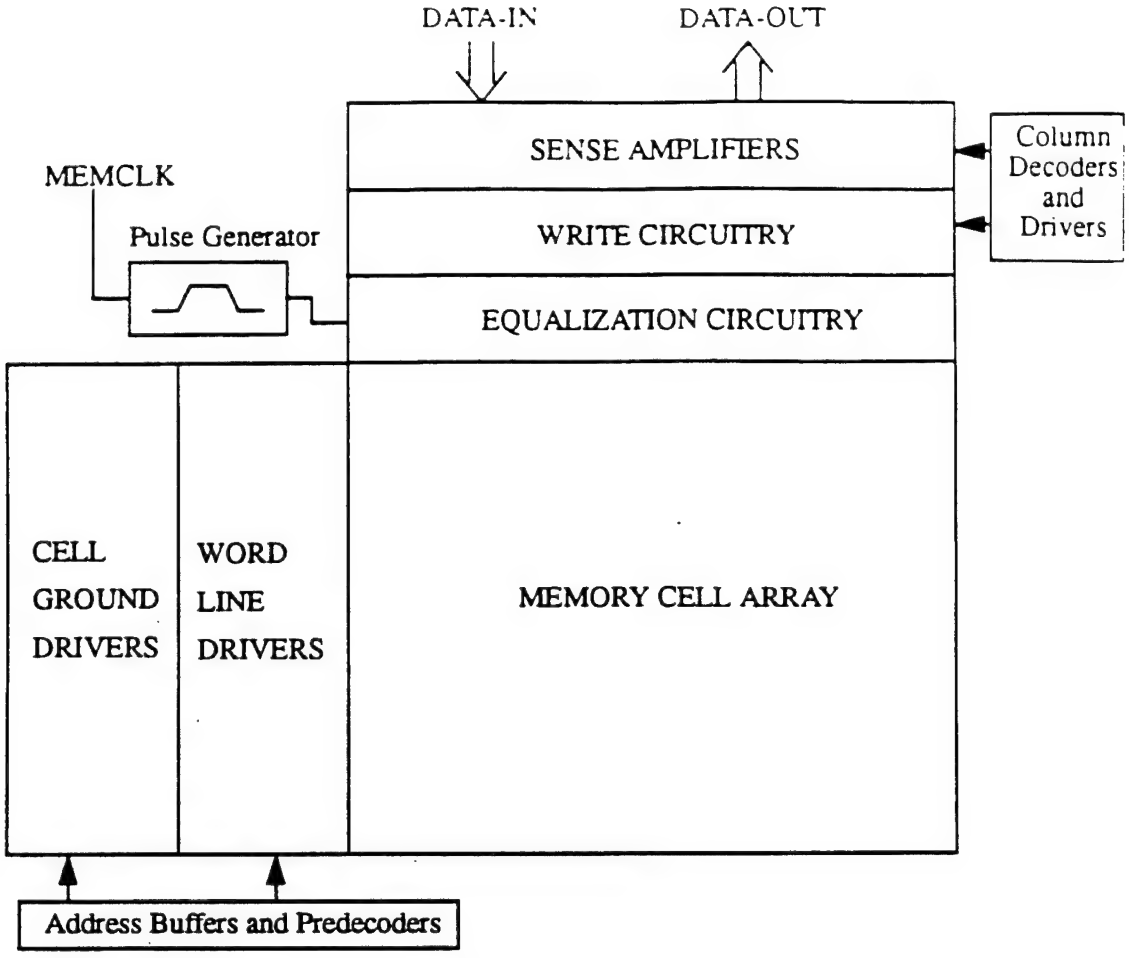


Fig. 5.2: Block diagram of the RAM layout

drivers have been tiled such that routing is achieved by simply abutting the cells to each other. The row and column address buffers and predecoders are made using standard cells and are automatically routed using automated place-and-route tools.

In the following sections the major circuits used in the RAM are described.

5.2.2 The Memory Cell

The memory cell used in the compiler is the CMMC. It is shown again in Fig. 5.3. During the read operation, the cell-ground is maintained at its standby bias of about 0V. The word line is brought low toward ground so that the access transistors *MA1* and *MA2* appear as current mirrors to the memory cell driver transistors. The static power dissipation of the cell, the read and write times, and the area of the memory depend on the size of the pullup transistors.

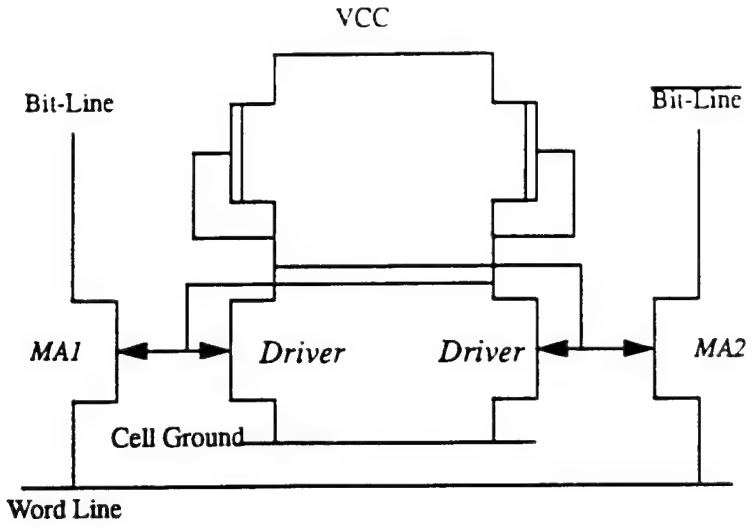


Fig. 5.3: The current-mirror memory cell

The drain-source read access current of the access transistor is given by

$$I_{DS} = K \cdot (V_{GS} - V_{TH})^2 \cdot (1 + b(V_{GS} - V_{TH})^{-1}) \cdot (1 + \lambda V_{DS}) \cdot \tanh(\alpha V_{DS}) \tag{5.1}$$

where K is the transistor transconductance which scales to first order with the W/L ratio of the device, V_{TH} is the threshold voltage, b is a velocity saturation parameter, λ is a channel length modulation parameter, and α is the drain voltage multiplier. An increase in load transistor current logarithmically increases the access transistor gate-source voltage. This increases the read-access current quadratically, as seen in (5.1).

The write time is a function of the access transistor gate-drain diode resistance. This resistance is given by

$$r_{DIODE} \approx \frac{2nkT}{qI_{CELL}} \tag{5.2}$$

where T is the temperature, k is Boltzman's constant, n is the ideality factor of the diode and I_{cell} is the total cell current. A smaller write diode resistance leads to a faster write time.

The read access time, the write time, and the cell area are reduced at the expense of increased power dissipation by reducing the length of the pullup devices. Since the choice of cell pullup transistor length has a dramatic impact on speed, power and area, the cell has been designed to allow a range of sizes for the pullup transistors.

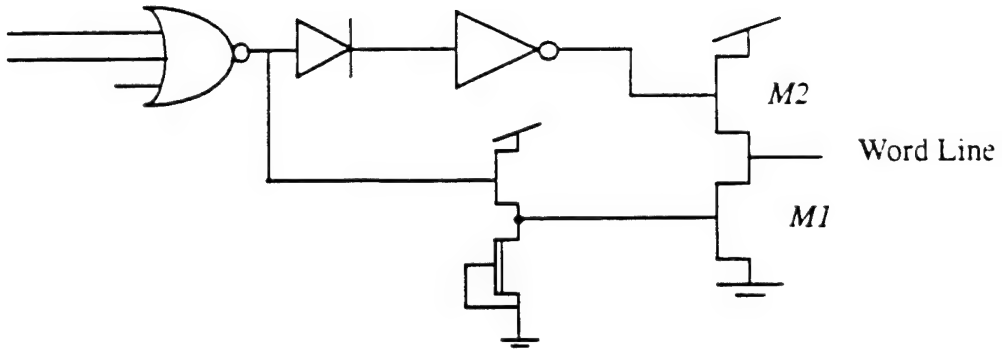


Fig. 5.4: Word line driver

The access transistor width also has a significant impact on the speed and the area of the memory. For this reason, the access transistors have also been designed to allow for a large range of sizes. The peripheral circuitry has been designed to adjust with the size of the cell to remain pitch-matched to the cell while utilizing the area efficiently.

5.2.3 Word Line Driver

The word line driver is shown in Fig. 5.4. The input NOR, which combines predecoded address signals with a read signal, is fixed at minimum size to minimize the diode load on the read and predecode lines. The source follower is sized to be the geometric mean of the input and output transistor sizes, thus achieving uniform scaling within the cell. The size of the output pull-down transistor, *M1*, controls the word-line low voltage. The low voltage affects the memory cell access transistor bias according to the equation

$$V_{GS}(access) = V_{CS} + V_{CG} - V_{WORD} \tag{5.3}$$

where V_{CS} is the cell storage voltage (with respect to cell ground), V_{CG} is the cell ground voltage and V_{WORD} is the word line voltage at the cell. As the size of M1 increases, the word-line low voltage decreases, which increases the memory cell access transistor gate-source voltage. As a result, the read-access current is increased and the access time is reduced. The trade-off is that a larger pulldown transistor *M1* achieves a faster access time at the expense of an increase in the overall area of the memory.

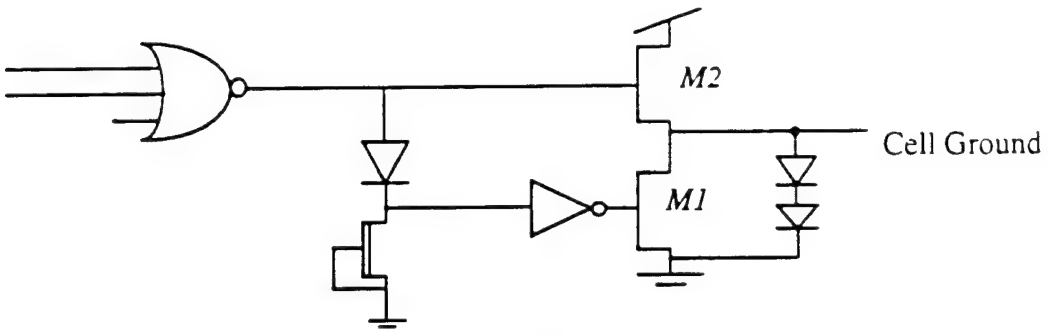


Fig. 5.5: Cell-ground driver

During the read operation, the word lines of nonselected rows are raised, shutting off the access transistors in these rows by reducing their gate-source (storage node-to-bit line) voltage. The word-line high voltage is determined by the size of the word-line pullup transistor, *M2*, the leakage across the pulldown transistor, *M1*, and the strength of the inverter driving *M2*. This voltage is critical in determining the suppression of leakage currents which flow from the bit-lines through the access transistors in rows that are not selected. The amount of leakage current suppression can be determined by the reverse bias applied to the memory cell access transistor when it is turned off. This margin is given by:

$$V_{rev\_bias} = V_{word\_hi} - V_{cell\_storage}$$

$$= V_{word\_hi} - (V_{ch} + V_{cg\_lo}) \tag{5.4}$$

where $V_{cell\text{-}storage}$ is the cell storage node voltage, V_{ch} is the cell high voltage with respect to cell-ground, $V_{cg\text{-}lo}$ is the cell-ground low voltage. Transistor measurements indicate that a reverse bias of 0.3V achieves the maximum suppression of leakage current. The noise margin can be increased from within the word-line driver by increasing the width of *M2*.

The word-line buffer in the compiler thus allows arbitrary scaling of both the output pull-down transistor, *M1*, to control the read time, and the pullup transistor, *M2*, to control the leakage current margin.

5.2.4 Cell-Ground Driver

The cell-ground driver, shown in Fig. 5.5, controls the ground line of the cells in a row. The driver consists of a NOR gate (which combines the predecoded address signals and a write

signal), an inverter, and a push-pull driver loaded with a series chain of two diodes to ground. As with the word-line driver, the NOR gate of the cell-ground driver is kept minimum in size to minimize the gate load on the predecode and write lines.

During the read operation, the cell-ground voltage remains at its standby low voltage near ground. This low voltage is controlled by the size of the pulldown transistor, *M1*, which has conflicting requirements. A faster readout can be achieved with a higher cell-ground low voltage, as in (5.3), by using a smaller size for M1. Leakage currents can be minimized with as low a cell-ground low voltage as possible, as shown by (5.4), by using a larger value for the size of M1.

The cell-ground pull-down transistor also affects the write time. The cell-ground voltage must be raised by at least one diode drop above ground to write to the cell. By starting with a higher cell-ground low voltage, less time will be required to raise the cell-ground voltage, resulting in a faster write time. Raising the cell-ground low voltage arbitrarily, however, can be dangerous. If the cell-ground low voltage is too high, it is possible to inadvertently write to cells in the wrong row.

The size of the cell-ground output pullup transistor, *M2*, determines how quickly the cell-ground voltage will be raised, and hence how quickly the write operation can occur. The trade-off, again, is one of speed versus area.

The cell ground output pullup transistor and pulldown transistors are allowed to scale arbitrarily to allow control of the write time, leakage current noise margin, and read time.

5.2.5 Pulse Driver

An integrated pulse-generator and a driver are used to produce a self-timed pulse from the synchronous clock-signal, MEMCLK. At the beginning of a cycle, the bit line voltages may be spread quite far apart due to a previous read or write. Therefore, it is necessary to reset the bit lines so that they are ready for the current operation. The purpose of the equalization circuitry is to precharge both bit lines high, and to bring their potentials as close together as possible, and into the range where the sense-amplifier is most sensitive. The purpose of the pulse circuitry is to generate a pulse to control the equalization circuitry. This pulse must be long enough for the equalization circuitry to reset and release the bit lines by the time the new decoded address has reached the

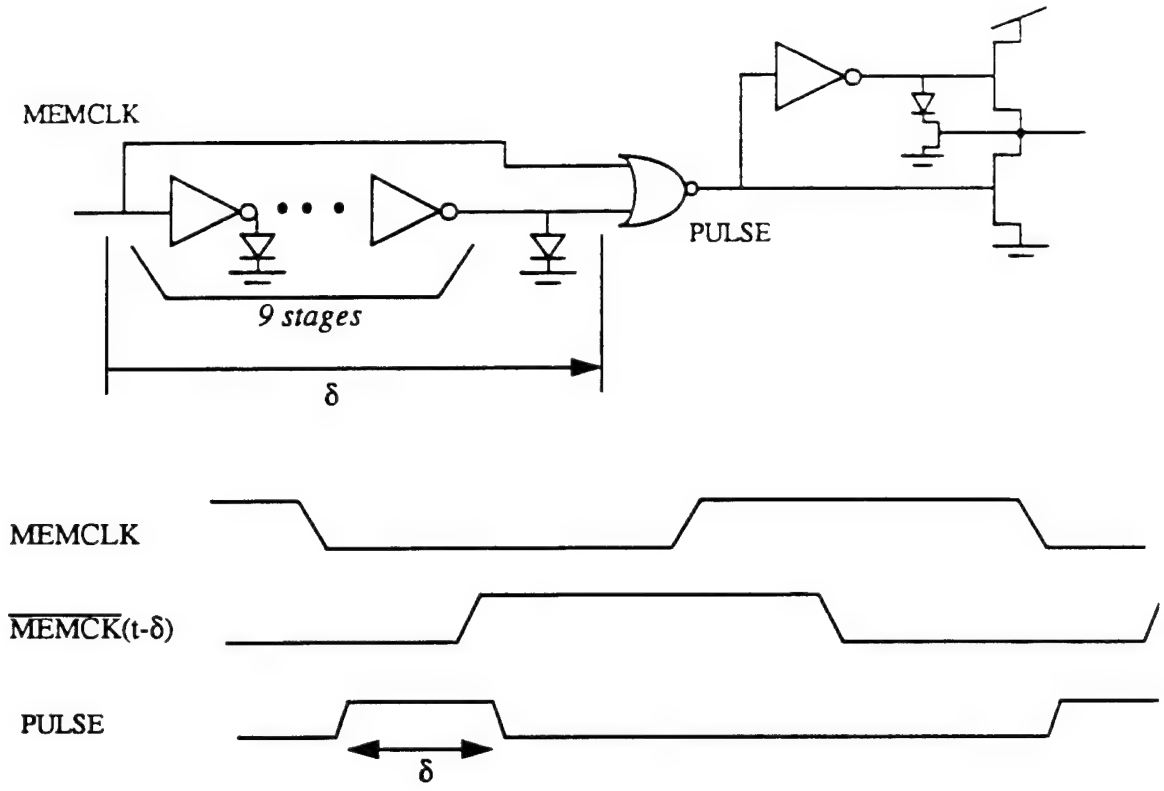


Fig. 5.6: Equalization pulse generator and timing diagram.

memory cells.

There are two basic requirements of the generated pulse width. The first is that it must be variable over the range needed for RAMs of different sizes and speeds. The second is that it must be well controlled across process variations.

The pulse generator shown in Fig. 5.6 satisfies these criteria. The generator is made of a NOR gate which has as inputs MEMCLK and a suitably delayed and inverted version of MEMCLK, $\overline{\text{MEMCLK}}(t-\delta)$. Fig. 5.6 also shows a timing diagram depicting the operation of the pulse generator. When the input MEMCLK is high, the inputs to the NOR gate are a 1 and a 0, and the PULSE signal is thus a 0. When MEMCLK is lowered, the inputs to the NOR gate are both 0s, producing a 1 on the PULSE output. This condition persists for a delay δ, in which time the transition in MEMCLK propagates through the inverter chain. MEMCLK must be active low for a duration at least equal to the inverter chain delay, δ, to prevent pulse swallowing.

The delay of a DCFL circuit is a strong function of fanout. Hence, to control the delay of

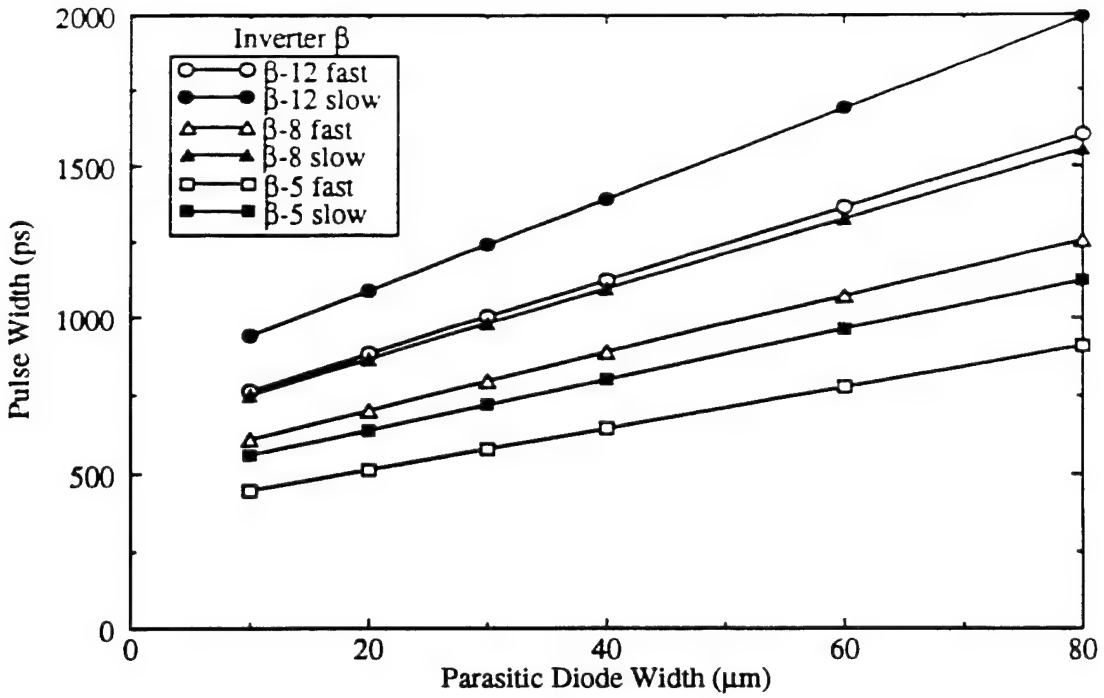


Fig. 5.7: Impact of the pulse generator parasitic diode loads and inverter β ratio on pulse width.

the inverter chain, variable-sized diodes have been added to each node in the chain. To increase the granularity of this control, the strengths of the inverter pullups in the chain are also variable. Fig. 5.7 shows the effects of the load diode width and inverter β on the resultant pulse width. The range of delays expected across process variations from the fast-fast to slow-slow corners are shown in the figure.

Optimal pulse generator transistor sizes can be found in a two-step procedure. In the first step, the diode size is fixed in the middle of the allowed range and an inverter β ratio is found which gives a pulse that produces the best access and write times. In the second step, the diode load is varied using this β to achieve the best performance. Since a particular pulse width can be generated using several different combinations of β value and diode load, this method is guaranteed to find an optimal pulse width.

Fig. 5.8 shows the variation in pulse width across process variations for different inverter β values. This figure shows that the pulse width control is independent of β.

5.2.6 Equalization Circuitry

The equalization circuitry is shown in Fig. 5.9. It precharges the bit lines high and brings

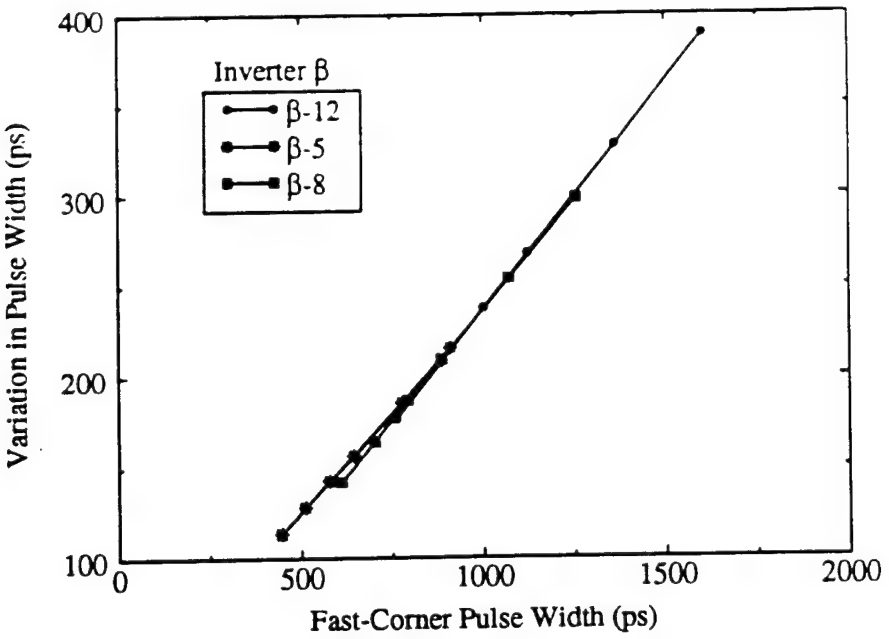


Fig. 5.8: Impact of inverter β on process-induced variation on pulse width.

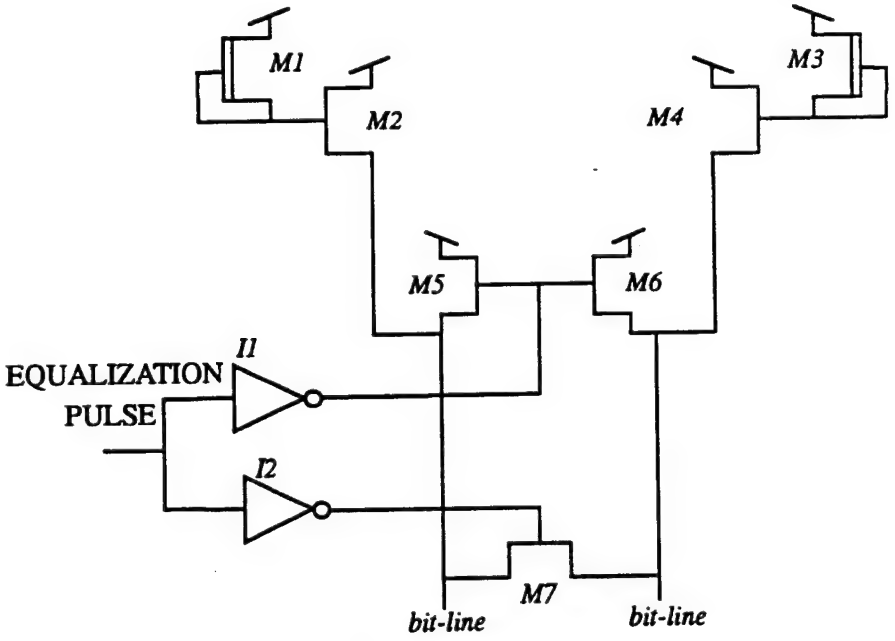


Fig. 5.9: Equalization circuitry schematic

their potentials close together in a range where the sense-amplifier is most sensitive. This operation must be performed quickly so that it does not increase the length of the critical path.

Transistors *M1* through *M4* are used to provide static pullup for the bit-lines. The remainder of the equalization circuitry is controlled by the pulse generator. The precharge operation is

achieved using inverter *I1* and transistors *M5* and *M6*. Inverter *I2* and transistor *M7* are used to bring the bit line potentials together.

To achieve sub-1.5ns equalization times, fixed sizes were found for transistors *M5*, *M6*, and *M7*. The supply current that is drawn through transistors *M5* and *M6* during the precharge operation is a function of the sizes of *M5* and *M6*, the bit-line separation, and the bit-line capacitance. Inverters *I1* and *I2* always draw supply current, and can be sized independently sized to trade power and area for speed as required.

5.2.7 Sense-Amplifier

The sense-amplifier, shown in Fig. 5.10, consists of 3 stages. The first stage presents a low-impedance path from the bit-lines to ground. To minimize access transistor leakage currents, the memory cell requires the bit-lines to be biased in the 1.0 to 1.4V range; level shifting diodes D1 and D2 accomplish this.

The bit-lines are connected to the equalization circuitry, the memory cell, the write circuitry, and the sense-amplifiers. Each of these circuits has some impact on the common-mode voltage of the bit-lines. The latching transistors *M3* and *M4* can be scaled to allow the first stage of

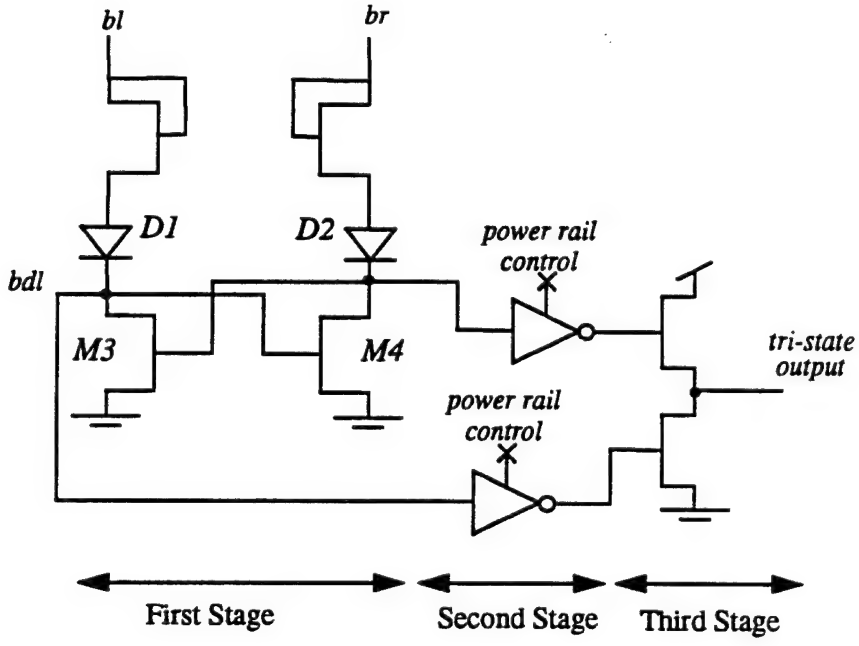


Fig. 5.10: Sense-amplifier schematic.

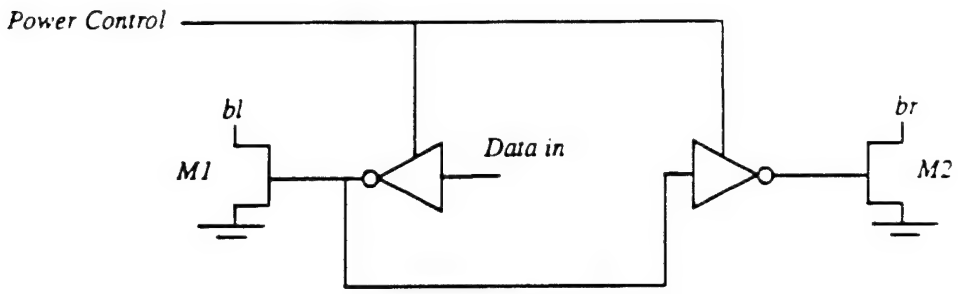


Fig. 5.11: Write circuitry schematic.

the sense-amplifier to adapt to different common mode voltages on the bit-lines. The translated and amplified bit-line voltage swing can be centered around the high-gain region of the second stage of the sense-amp by varying the size of these transistors.

The second stage of the sense amplifier uses two Power Rail Logic (PRL) inverters to drive a tristate bus. The power rail of these inverters is used as a column select input. When the power rail control input is raised, the second stage of the sense amplifier can pass its output to the tri-state driver. When the power rail control input is lowered, the outputs of the PRL inverters are brought to ground, removing the sense amplifier outputs from the bus. In the degenerate case with no column folding, the power control signals can all be tied directly to the supply voltage.

5.2.8 Write Circuitry

Like the read circuitry, the write circuit (shown in Fig. 5.11) uses power rail logic. The power control signal acts as a column select signal as well as a write enable signal. When the write control line is raised, data is written to the bit-lines. When this signal is lowered, the output of both inverters are forced low, turning off write devices *M1* and *M2*.

By varying the sizes of *M1* and *M2*, control over the write time can be exercised. The trade-off is that larger write transistors achieve faster write times at the expense of area. The power-rail inverters are sized to maintain adequate power rail high-voltage noise margins, while the write transistors, *M1* and *M2* are scalable.

5.2.9 Read/Write, Address and Predecode Buffers

The address buffers and predecoders are made using feedback-FET logic buffers, as

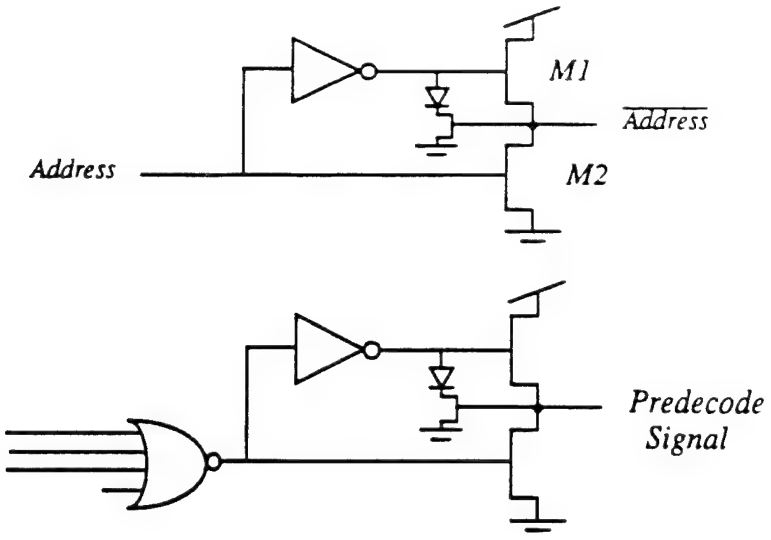


Fig. 5.12: Address buffer and predecode buffer schematics

shown in Fig. 5.12. Automatic buffer sizing capabilities are provided for both types of cells. The inverters in these buffers can be minimum size because of relaxed signal swing requirements on their output nodes.

In the address and read/write buffers, the output transistors, *M1* and *M2*, are scalable and are of equal size. For the predecode buffer cell, the strength of the NOR gate and the output transistors are also scalable over a large range of sizes. By simply making these devices larger, the delay and output slew rates (which affect the delay of successive stages) are reduced. The automatic transistor size determination program described in Chapter 4 was used to find a suitable range of transistor sizes to efficiently drive the buffer's expected loads.

Fig. 5.13 is a block diagram showing the readout and write path with all the components described in this section.

5.3 Compiler Structure

Fig. 5.14 is a flow-chart showing the organization of the RAM compiler. The compiler takes a Verilog description of the RAM as input. This description includes a specification of the number of rows and columns in the RAM and the number of address bits. Two additional parameters define how power rail logic is used in the readout and write circuitry.

The user must specify a target read access time and a target write time. Based upon the

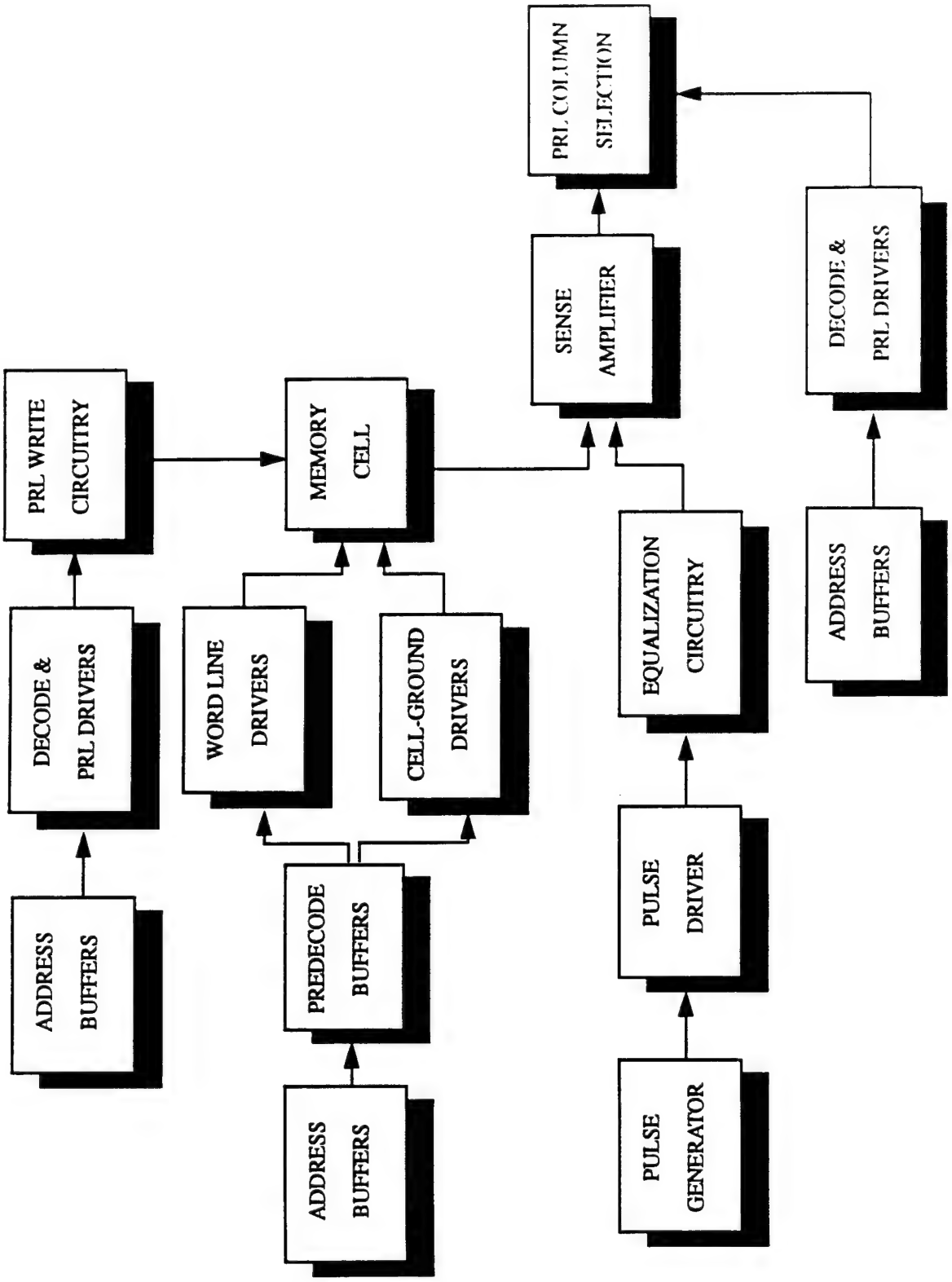


Fig. 5.13: Block diagram of the readout and write paths.

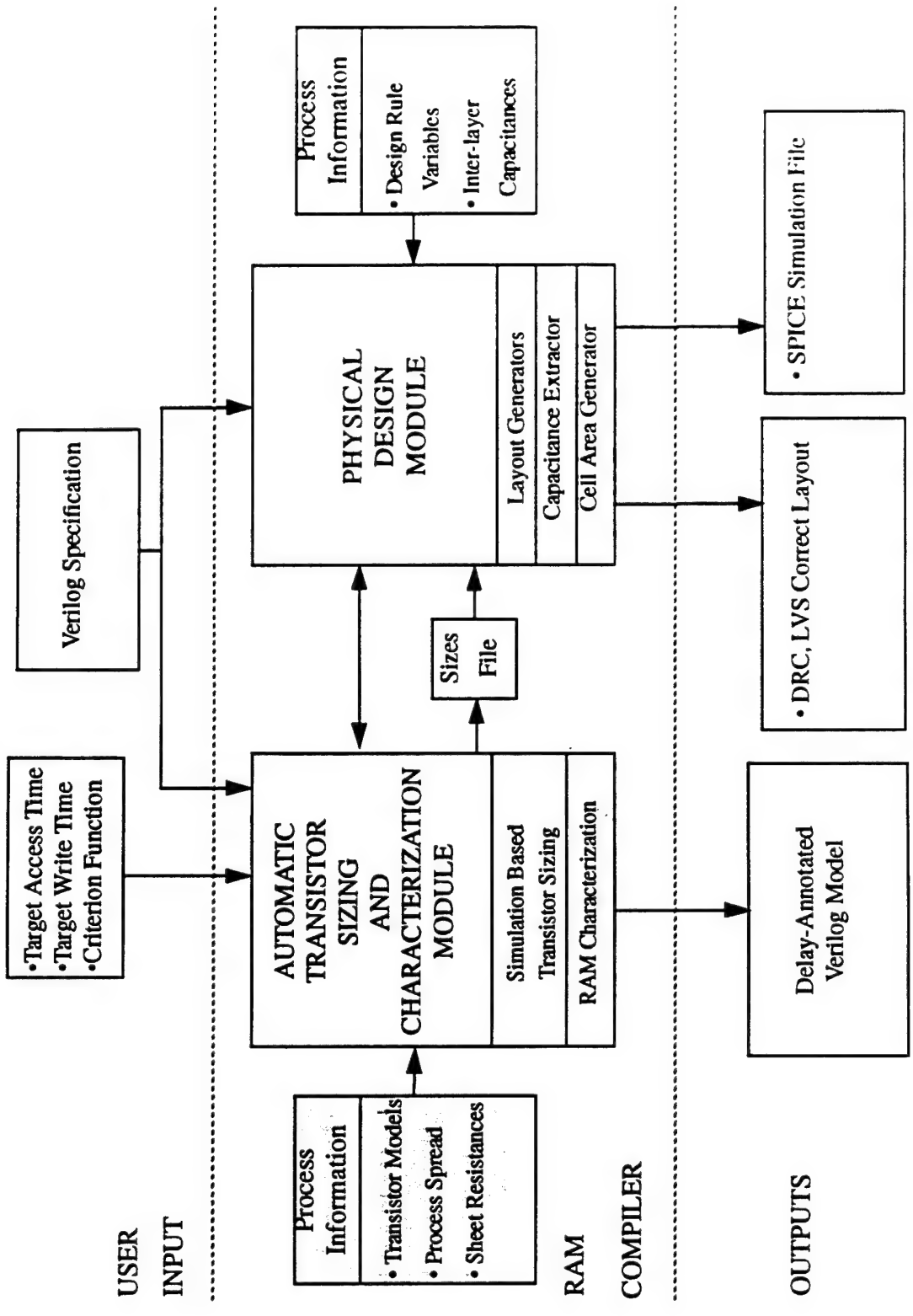


Fig. 5.14: Compiler structure flowchart.

circuit design discussion in the previous section, there are many different ways to improve the read access and write times, each with different costs in area and power dissipation. The user has the ability to specify whether power dissipation, area, or the time required to find a solution should be used to guide the transistor sizing procedure.

The RAM compiler consists of two software modules, the physical design module and the transistor sizing module. Together, these modules form a framework that can readily adapt with changes in processing technology. This flexibility is illustrated by the shaded boxes in the flow chart which denote process definition files.

The transistor sizing module, for instance, takes the transistor HSPICE models and process spread information as inputs. Since the transistor sizing and RAM characterization rely on this information, the compiler can re-construct memories in a different processing technology by simply using a new transistor model file or a new process spread file.

As a second example of its flexibility, the compiler performs iterative parameter extraction and transistor sizing to achieve desired delay goals. The extraction is performed using an inter-layer capacitance file and a metallization sheet-resistance file. Thus, one can adapt transistor sizing decisions, delay calculations, and power dissipation calculations to process changes such as different metallization thicknesses, or changes in dielectrics, by updating the associated process-files.

The compiler produces three outputs. The first output is a design-rule-correct and layout-versus-schematic-correct layout of the RAM. The second output is a SPICE netlist for the RAM for further verification of the RAM. The third output is a delay-annotated Verilog file containing the read and write times, and address and data setup and hold times. In the following sections, the physical design module and the transistor sizing module are described in greater detail.

5.4 Physical Design Module

The physical design module consists of a layout generator, a capacitance extractor and a cell area calculator. These three programs take a Verilog description and a transistor sizes file as inputs.

5.4.1 Layout Generators

The layout generators were written using CDA's Compiler Development System (CDS). CDS generates cells with variable transistor sizes. Additionally, CDS offers enough design-rule independence for the compiler to automatically generate layouts with a new set of design rules. Finally, this system allows a seamless interface for the integration of compiled SRAMs and other compiled circuits.

The block diagram of Fig. 5.2 shows the main components of the SRAM. The layout consists of one main SRAM block and a group of standard cells used for the row and column address buffers and predecoders. The main block was generated by hierarchically tiling an array of memory cells, cell-ground and word-line drivers, sense amplifiers, write circuitry, equalization circuitry, and pulse drivers.

A netlist, describing the interconnection of address buffers and predecode cells, was used to generate a group of standard cells. The user can automatically or manually place this group of standard cells. After placement, the compiler automatically routes the address buffers and predecode cells to the main SRAM block.

5.4.2 Transistor Sizing

An important feature of the layout generators is that each cell includes provisions for scaling transistor sizes. This allows the compiler to use buffer sizes that are appropriate for the size of the generated SRAM. Fig. 5.15 shows two examples of the memory cell layout. All transistors in the cell are individually scalable. The layout on the left (Fig. 5.15 (a)) shows a cell using the minimum sizes allowed by the compiler. The cell on the right (Fig. 5.15 (b)) shows the same cell using a longer pullup device and a wider access transistor size. Although the second cell consumes less than 50% as much power, it occupies 38% more area.

Fig. 5.16 (a) and 5.16 (b) show word line drivers that were generated to pitch-match with the memory cells in Fig. 5.15. In this example, the layout generator for the word-line driver took advantage of the increased height of the second memory cell by decreasing the width of the driver cell. This was accomplished by using longer transistor fingers and less transistor folding on the output transistors.

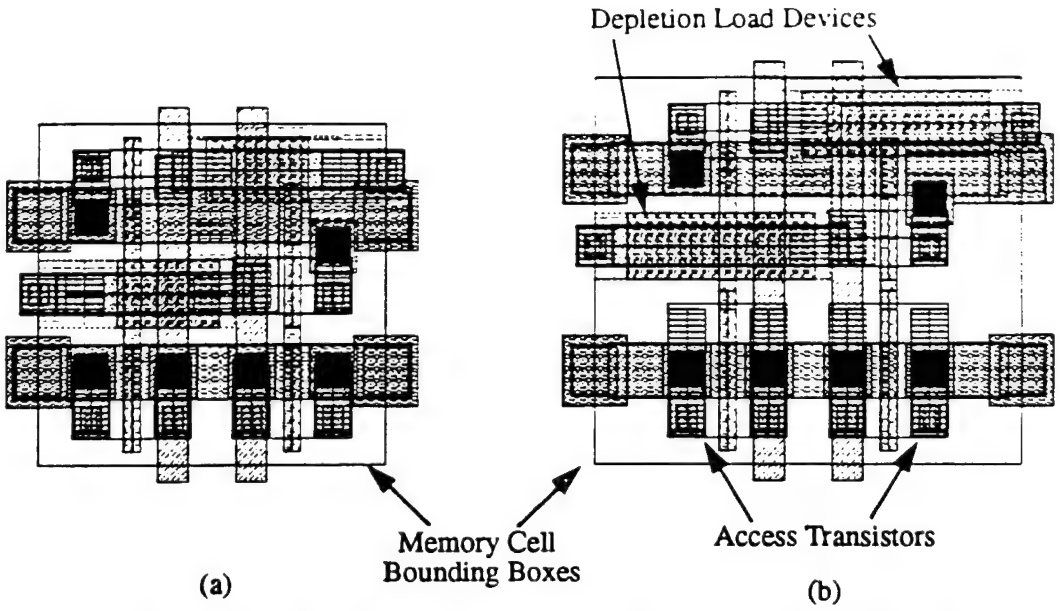


Fig. 5.15: Examples from the scalable 1-read, 1-write memory cell layout generator.

5.4.3 Capacitance Extraction

Capacitances on nodes that are shared across cells, and capacitances on nodes that are local to a cell, are treated differently by the compiler. A program that interfaces to the CDA tools

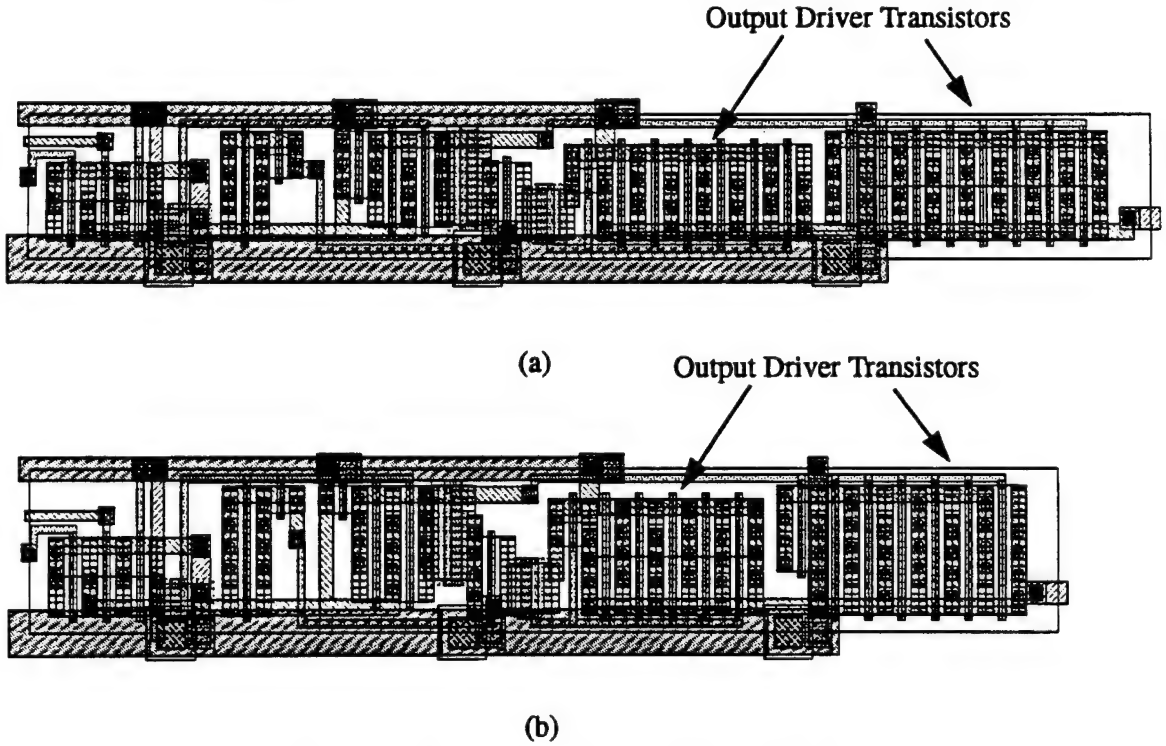


Fig. 5.16: Examples of the scalable word-driver.
(a) word driver pitch-matched to the memory cell in Fig. 5.15(a);
(b) word driver pitch-matched to the memory cell in Fig. 5.15(b).

is used to extract the capacitances of the bit-lines, word and cell-ground lines, predecode address lines, and a handful of control signals that are shared across cells. These capacitances can be accurately determined by performing capacitance extraction on only a few cells. Capacitances on nodes within a cell are dynamically computed during simulation based on transistor dimensions within the cell. This is identical to the dynamic capacitance calculation described in Section 4.2.

5.4.4 Cell Area Determination

The final component of the physical design module is a program that determines cell area. This program takes as input the transistor sizes and organization of the SRAM. The dimensions of individual cells and of the overall SRAM array are calculated using the design-rule independent variables, the transistor sizes, and the power-rail sizes. This process is very rapid and does not require any layout generation.

The cell dimensions are used in two different ways by the transistor sizing module. First, the dimensions are used to permit resistance extraction on all critical nets. Second, cell dimensions are used to determine the area penalty associated with different transistor sizing choices. Thus, area can be used as a cost function for guiding transistor size selection.

5.5 Transistor Sizing Module

The transistor sizing module consists of a collection of programs written in the Perl scripting language. This module is a simulation engine that methodically sizes transistors to meet a user-specified access and write time. In the search for optimal transistor sizes, the physical design module is used to generate layout and to extract parasitic capacitances; HSPICE is used to perform circuit simulation, and Perl scripts are used to analyze the results of the circuit simulations. In addition to performing automatic transistor sizing, this module also characterizes the generated SRAM.

An approach commonly used in CMOS SRAM compilers is to develop and to employ lookup tables or macromodels for calculating delays and power dissipations. An advantage of this approach is that calculations are orders of magnitude faster than performing simulations using a circuit simulator such as HSPICE. Another advantage is that SRAM compilers which are devel-

oped as part of a larger CAD framework can make extensive use of available lookup tables and macromodels.

There are, however, three major cost-disadvantages to this approach. The first drawback is the significant cost associated with macromodel development. Once the macromodels are developed, they are tied to specific circuit structures designed in a specific process.

The second drawback is a high maintenance cost. In E/D MESFET GaAs circuit design, small noise margins, leakage currents, and process variations make it necessary to develop accurate macromodels for calculating delays *and* signal levels to ensure signal integrity in critical portions of the RAM. Although accuracy can be achieved by embedding circuit equations into the compiler, the cost of developing and maintaining such a transistor-level circuit simulator is very high. As new transistor equations are developed to cope with the ever-shrinking channel lengths, the long-term cost associated with maintaining a circuit simulator specifically for the compiler becomes too large.

The final cost-disadvantage is also related to high maintenance. The major disadvantage of using lookup tables or macromodels is that they are only valid at a particular process corner. The cost of generating and maintaining lookup tables or macromodels to model the effect of process variations in a RAM quickly becomes excessive as the efforts must be repeated for each process corner.

One advantage of using a circuit simulator such as HSPICE to perform the required calculations and optimizations is the assurance of relatively accurate delay and power calculations. Another important advantage is that no cost is associated with developing or maintaining macromodels or lookup tables. The implication is that a framework developed by this approach can readily adapt to changes in processing technology by simply updating a transistor models file used by the simulator.

The most important advantage of using a circuit simulator in the SRAM compiler is that signal levels and noise margins can be determined very accurately and easily across process spreads and operating conditions. This advantage is critical for the development of GaAs E/D MESFET SRAMs, and is therefore the approach that was taken for delay modeling, power estimation, and signal evaluations in the ARC.

The use of HSPICE as a simulation engine has enabled the development of an extremely powerful approach that permits a high degree of design verification. In the sections that follow, the verification that has been enabled across operating sequences and process variations will become evident.

5.6 Transistor Sizing Problem Definition

The transistor size selection problem can be defined as follows. A set of n transistor sizes, $x_1,x_2,..,x_n$, is sought that will minimize an objective function, f, of these n sizes, subject to a finite number of constraints. Formally, this can be written as a search for

$$min \cdot f(\boldsymbol{x})$$

where

$$f(\boldsymbol{x}) = P(\boldsymbol{x}), \qquad \text{(O-1)}$$

$$f(\boldsymbol{x}) = A(\boldsymbol{x}), \text{ or} \qquad \text{(O-2)}$$

subject to

$$t(\boldsymbol{x}) = min(t_{access}, t_{write}) \leq t_{target} \qquad \text{(C-0)}$$

where $\boldsymbol{x}$ is the vector of sizes $x_1,x_2,..,x_n$, $P(\boldsymbol{x})$ is the power dissipated by the RAM, $A(\boldsymbol{x})$ is the area of the RAM, $t(\boldsymbol{x})$ is the larger of the access time and the write time, and t_{target} is the target clock period for the SRAM.

The solution to the first objective function, O-1, represents a RAM that will achieve the desired access and write times while minimizing the power-delay product. Similarly, a solution to the second objective function will achieve an SRAM that meets the target access and write times while minimizing its delay-area product.

To guarantee that the vector of n sizes, $\boldsymbol{x}$, will produce a process tolerant and functional design, the objective function needs to be minimized with respect to several additional constraints. Before introducing these constraints, some terminology is defined.

The operation of successfully writing data D to a cell in row i and column j, given a previous operation or condition Y, shall be written as

$$W_D(C_{i,j}) \mid Y$$

The write operation is said to be successful if the cell is written correctly and all cells not being written maintain their state. Similarly, the output of a sense amplifier from a read operation of data D from a cell in row i and column j, given a previous operation or condition Y, shall be written as

$$R_D(C_{i,j}) \mid Y \quad .$$

To achieve process-tolerant design, functionality constraints must be met at each process corner that defines the process space. Thus, if our process space is defined by m process corners, and we have n different functionality constraints, the total number of constraints that must be satisfied to produce a process tolerant functional design is m x n.

Using the notation defined above, the functional constraints, $g_{n,k}(x)$, that must be met by the SRAM can be stated as

$$g_{1,k}(x) = W_0(C_{i,j}) \mid C_{i,j} = 0 \tag{C-1}$$

$$g_{2,k}(x) = W_1(C_{i,j}) \mid C_{i,j} = 0 \tag{C-2}$$

$$g_{3,k}(x) = W_1(C_{i,j}) \mid C_{i,j} = 1 \tag{C-3}$$

$$g_{4,k}(x) = W_0(C_{i,j}) \mid C_{i,j} = 1 \tag{C-4}$$

$$g_{5,k}(x) = R_1(C_{i,j}) \mid W_0(C_{z,j}) \geq V_{IH}, \forall\, z \neq i \tag{C-5}$$

$$g_{6,k}(x) = R_0(C_{i,j}) \mid W_1(C_{z,j}) \leq V_{IL}, \forall\, z \neq i \tag{C-6}$$

$$g_{7,k}(x) = R_1(C_{i,j}) \mid W_1(C_{z,j}) \geq V_{IH} \tag{C-7}$$

$$g_{8,k}(x) = R_0(C_{i,j}) \mid W_0(C_{z,j}) \leq V_{IL} \tag{C-8}$$

$$g_{9,k}(x) = R_0(C_{i,j}) \mid R_1(C_{z,j}) \leq V_{IL}, \forall\, z \neq i \tag{C-9}$$

$$g_{10,k}(x) = R_1(C_{i,j}) \mid R_0(C_{z,j}) \geq V_{IH}, \forall\, z \neq i \tag{C-10}$$

$$g_{11,k}(x) = W_0(C_{i,j}) \mid \{C_{i,j} = 1, C_{z,j} = 0\} \quad , \forall\, z \neq i \tag{C-11}$$

$$g_{12,k}(x) = W_1(C_{i,j}) \mid \{C_{i,j} = 0, C_{z,j} = 1\} \quad , \forall\, z \neq i \tag{C-12}$$

$$g_{13,k}(x) = W_1(C_{i,j}) \mid \{C_{i,j} = 0, C_{z,j} = 0\} \quad , \forall\, z \neq i \tag{C-13}$$

$$g_{14,k}(x) = W_0(C_{i,j}) \mid \{C_{i,j} = 1, C_{z,j} = 1\} \quad , \forall z \neq i \tag{C-14}$$

$$g_{15,k}(x) = V_{revbias} = V_{word} - V_{cell,storage} > leak \tag{C-15}$$

$$g_{16,k}(x) = x_{min} \leq x \leq x_{max} \tag{C-16}$$

These 16 constraints are applied at each of the k=1,2,...,m process corners that define the process space.

The first four constraints state that the write operation must be able to store the proper value in a cell regardless of the previous value in that cell.

The fifth and sixth constraints ensure that a read operation is successful when the read occurs immediately after writing the opposite data to another cell in the same column. The inequality in these constraints ensures that the sense amplifier behaves functionally and that the output voltage of the sense amplifier meets minimum noise margin requirements.

The seventh and eighth constraints ensure that a read operation is successful when the read occurs immediately after writing the same data value to another cell in the same column.

Constraint numbers nine and ten verify that the sense amplifiers can successfully read either data value from a cell after having read the opposite value from a cell in another row of the same column.

Constraint numbers eleven through fourteen test functionality of the SRAM in the presence of the bit-line clamping effect for the CMMC that was described in Chapter 2. Due to this effect, the states of the memory cells within a column influence how low the bit lines can be pulled during a write. If all of the cells in a column store the same data value, then when the bit line associated with the high-voltage side is pulled down, it is prevented from being lowered all the way to ground.

Constraints eleven and twelve guarantee that if all of the data values stored in a column are the same, then a cell in that column can be written with the opposite data value. Constraints thirteen and fourteen assume the successful writing to a cell with the same data present in all of the other cells in the column.

Constraint fifteen represents a noise margin constraint. To minimize the leakage currents associated with nonselected rows during a read or write operation, the word lines of these rows are

raised to reverse bias the gate-source junctions of the access transistor MESFETs. Arbitrary scaling of the transistors in the memory cell, word-line driver, and cell-ground driver may reduce this reverse bias. To produce an SRAM that is tolerant of this known problem, constraint fifteen is used to ensure that the reverse bias achieved across process variations maintains a minimum, predetermined value. The *leak* parameter is used as a safety margin.

The sixteenth constraint insures dense *and* scalable layouts, by placing upper and lower bounds on the transistor sizes.

The transistor sizing problem has been cast as a classical non-linear optimization problem. A number of techniques could be used to solve such a problem. In the following section, various aspects of the memory modeling and simulation are discussed. This discussion will be followed by a description of the systematic approach that was used to solve the transistor sizing problem and to characterize the RAM.

5.7 Circuit Modeling and Simulation

The purpose of modeling the SRAM is to reduce the amount of time required for simulations without sacrificing accuracy. In CMOS SRAM compilers, a critical path consisting of one row and one column is typically extracted. Only one memory cell is needed for simulations since the effects of other cells on the bit lines and word lines can be modeled by bit-line and word-line capacitances.

5.7.1 The Circuit Model

In a GaAs CMMC based memory, there are many more effects that must be modeled than in CMOS, resulting in a larger and more complex circuit model. The modeling techniques that have been developed for this RAM are presented below in the order in which they appear in the readout path.

5.7.1.1 Address and Read/Write Buffers and Predecoders

The first elements along the readout path are the address buffers, predecode buffers, and read/write buffers. The outputs of these buffers are loaded by line capacitance and inputs to NOR

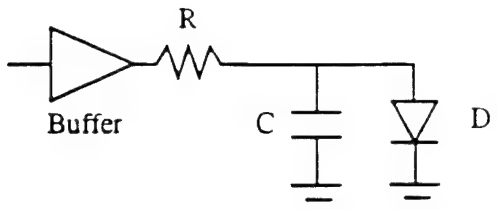


Fig. 5.17: Basic R-C-D model used to model the read/write, buffered address, and predecode line buffer loads.

gates. These loads are simply modeled by line resistances, lumped capacitances, and diodes that are wide enough to represent the total enhancement MESFET load on the lines. Fig. 5.17 shows this simple RCD model.

5.7.1.2 Word and Cell-Ground Drivers

The next set of buffers in the readout and write paths are the cell-ground and word-line drivers. Fig. 5.18(a) shows a simplified schematic of the cell-ground driver and its associated load. As discussed in Section 5.2.4, the precise cell-ground low voltage has a significant impact on noise margins, and on the read and write times. Since all n columns of memory cells in a row are driving equal amounts of current into the cell-ground driver, only one memory cell needs to be included in the model. The current from the remaining $n-1$ cells can be modeled using a dependent current source, $g = (n-1)*(I_1 + I_2)$, where I_1 and I_2 are the currents flowing through the source terminals of the driver transistors in the one memory cell. A schematic diagram showing this model is given in Fig. 5.18(b).

Similarly, a simplified representation of the word-line driver and its associated load is shown in Fig. 5.19(a). The access transistor currents affect the word line voltage, and must be accurately modeled. Only one cell in a row and a current-dependent current source, $g=(n-1)*(I_1 + I_2)$, are needed to model the effects of all of the access transistor currents flowing into the word-line. This model is shown in Fig. 5.19(b).

5.7.1.3 Resistive Drops

Resistive drops on the order of 100-150mV commonly occur along the word line. The cell current is typically much smaller than the current flow through the access transistors during read-

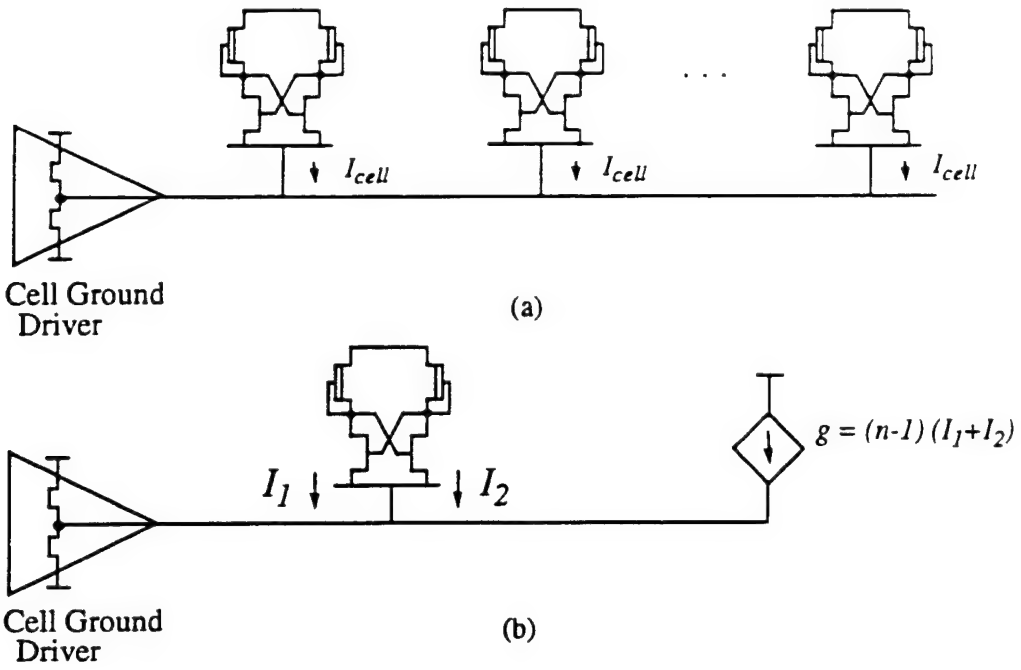


Fig. 5.18: Cell-ground driver loading and equivalent model.

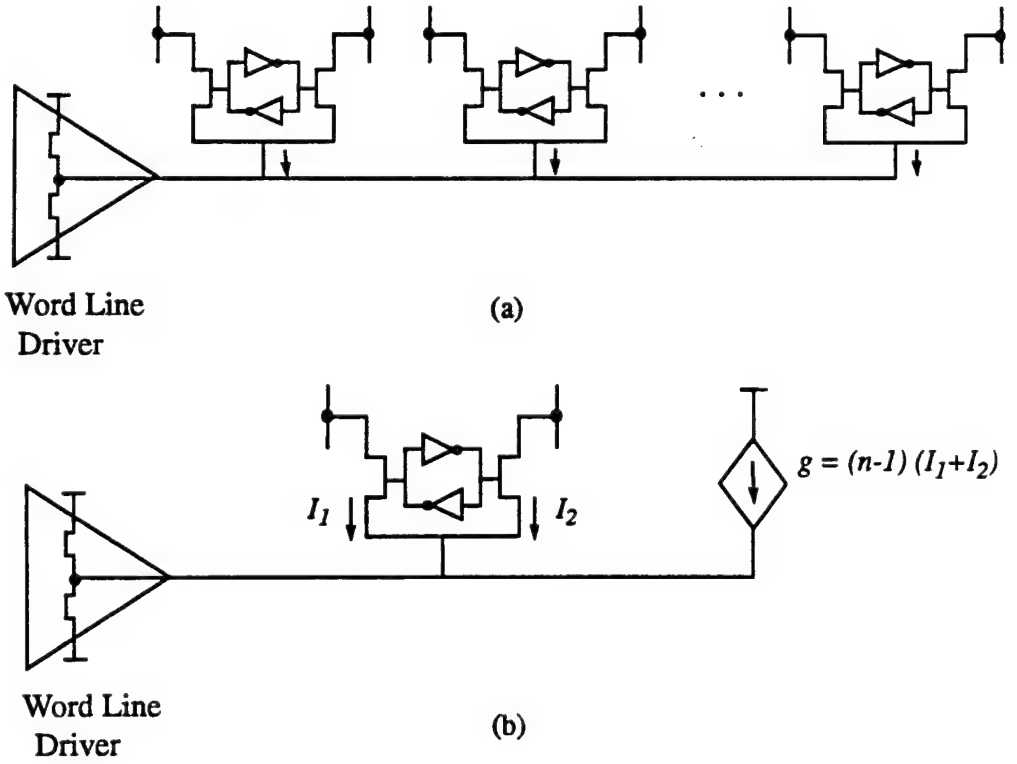


Fig. 5.19: Word-line driver loading and equivalent model.

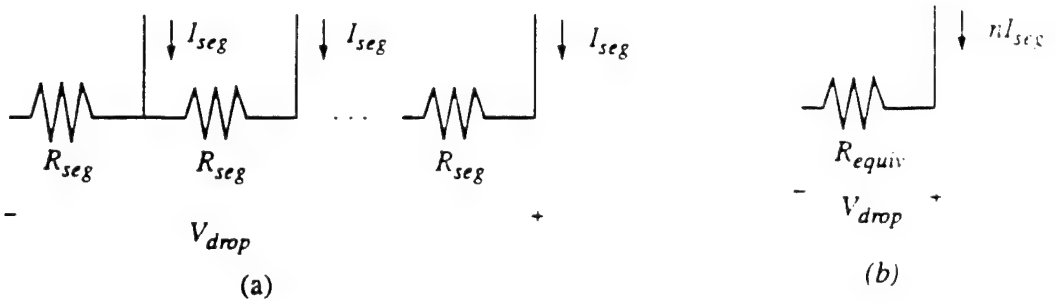


Fig. 5.19: Modeling the resistive drop due to a distributed current injection across a line. (a) distributed current injection; (b) equivalent model.

out. Although the resultant voltage drop across the cell-ground line is much smaller, it is also important.

The distributed I-R drops can be visualized using Fig. 5.20(a), where I_{seg} is the current injected at regular intervals along the line, and R_{seg} is the resistance of each segment. The total voltage drop across n segments of the line, V_{drop}, is given by:

$$V_{drop} = I_{seg}R_{seg} + 2I_{seg}R_{seg} + \cdots + nI_{seg}R_{seg}$$

$$= I_{seg}R_{seg}\sum_{i=1}^{n} i$$

$$= I_{seg}R_{seg}\frac{n \cdot (n+1)}{2}$$

This distributed voltage drop is modeled using an equivalent lumped resistor, as in Fig. 5.20(b), by equating the two expressions for V_{drop}.

$$V_{drop} = n \cdot I_{seg} \cdot R_{equiv}$$

$$= I_{seg}R_{seg}\frac{n \cdot (n+1)}{2}$$

giving

$$R_{equiv} = \frac{n+1}{2}R_{seg} \tag{5.5}$$

This equivalent resistance has been used to model the distributed I-R drop for both the cell-ground and word-lines.

5.7.1.4 Bit Line Effects

Constraint numbers eleven through fourteen test functionality of the SRAM in the presence of the bit-line clamping effect for the CMMC that was described in Chapter 2. The situation is described here using Fig. 5.21. If all of the cells in a column store the same data value, then the bit line is prevented from being lowered all the way to ground.

This effect needs to be properly modeled. A time-efficient solution is to add a dummy row to the simulation along with a pair of current-dependent current sources that model the current flow into the dummy row memory cell access transistors. The row is called a dummy row because it is never read or written; it is only initialized at the beginning of the simulation. By using dependent current sources on the bit-lines as shown in Fig. 5.22, the bit-line clamping can be precisely modeled. This model describes the worst-case scenario where all rows beside the active row of the simulation store the same data value as the dummy row.

5.7.1.5 RC Delay Modeling

The effect of RC delays in the predecode lines, word and cell-ground lines, bit lines, equalization pulse line, and PRL control lines is modeled. The impact of the I-R drops is critical

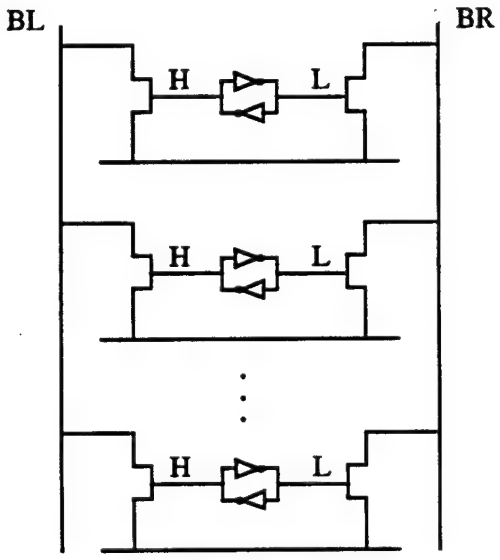


Fig. 5.20: Data-dependent bit-line clamping.
In this column, all of the storage nodes associated with BL are high (H).
When BL is brought low, the storage nodes limit how low the bit-lines can be pulled.

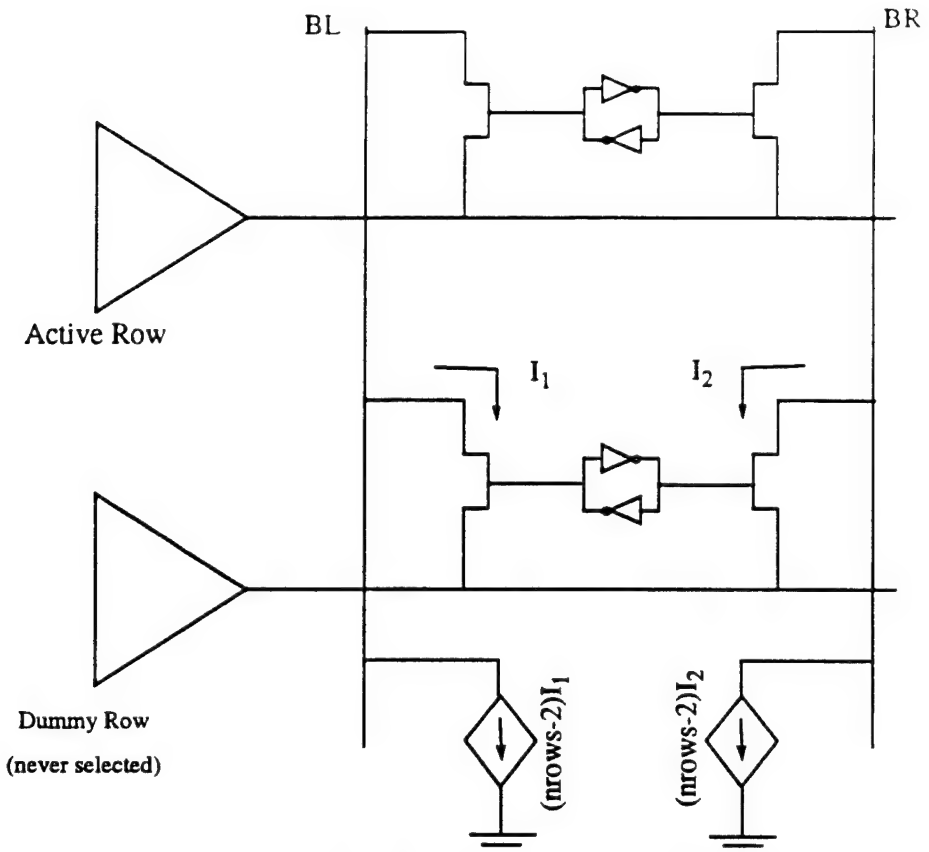


Fig. 5.21: A model of the data-dependent bit-line clamping.

for functionality, and hence the lumped resistance expression (5.5) is used for the word lines, cell-ground lines and PRL control lines. For the remaining signal lines, the I-R drop is not critical and therefore the total line resistance, nR_{seg}, is used to achieve a liberal estimate of the RC delay.

5.7.2 Dynamic Modeling

Various dynamic circuit modeling techniques are used to achieve accurate and efficient circuit models.

5.7.2.1 Transistor Sizes and Node Capacitances

Many of the transistor sizes can be scaled over a fixed range of values. Rather than requiring the individual cells to be re-extracted every time a transistor size is changed, dynamic capacitors are added to the nodes associated with these transistors. These dynamic capacitors have capacitances that are functions of transistor sizes, as in the modeling technique described in Sec-

tion 4.2.

5.7.2.2 Modeling Column Folding

Many embedded memory applications, such as instruction and data caches, require wide organizations to attain high on-chip bandwidth. Some memories require no column multiplexing (or folding). The amount of circuitry required for accurate modeling is dramatically reduced compared to what is required for an SRAM with column multiplexing. For a memory with an arbitrary amount of column folding, two columns are needed in the model — one active column that is being written, and one dummy column to ensure that data in nonselected columns is not written. For a memory requiring no column multiplexing, the dummy column is dynamically removed. This eliminates one column of memory cells, their associated dependent current sources, one sense-amplifier, one write cell, and one equalization cell from the model, reducing the simulation time by approximately 35%.

Fig. 5.23 shows a model of the SRAM with arbitrary column folding. The components enclosed in shaded boxes are removed when the circuit model is collapsed for a memory with only one column per bit. To simplify the diagram, resistive and capacitive components of the model are not shown.

5.7.2.3 Process Space Modeling

There are several sources of process variation in GaAs, such as nonuniform doping control and nonuniform linewidth control. As discussed in Chapter 3, the impact of these variations on circuit performance can be modeled, to a first order, by variations in threshold voltage. Most manufacturers of E/D MESFET GaAs circuits report [Wil94] that the threshold voltages of enhancement and depletion MESFETs tend to track each other so that process points lie in a band centered on a fast-fast to slow-slow process space line, as in Fig. 5.24.

Three different models are used in the compiler to describe this process space, as shown in Fig. 5.25. Each process model defines the process space by a number of process corners. These models include a two-point, three-point and six-point model.

The two-corner model includes the slow-slow and fast-fast corners. The coordinates of

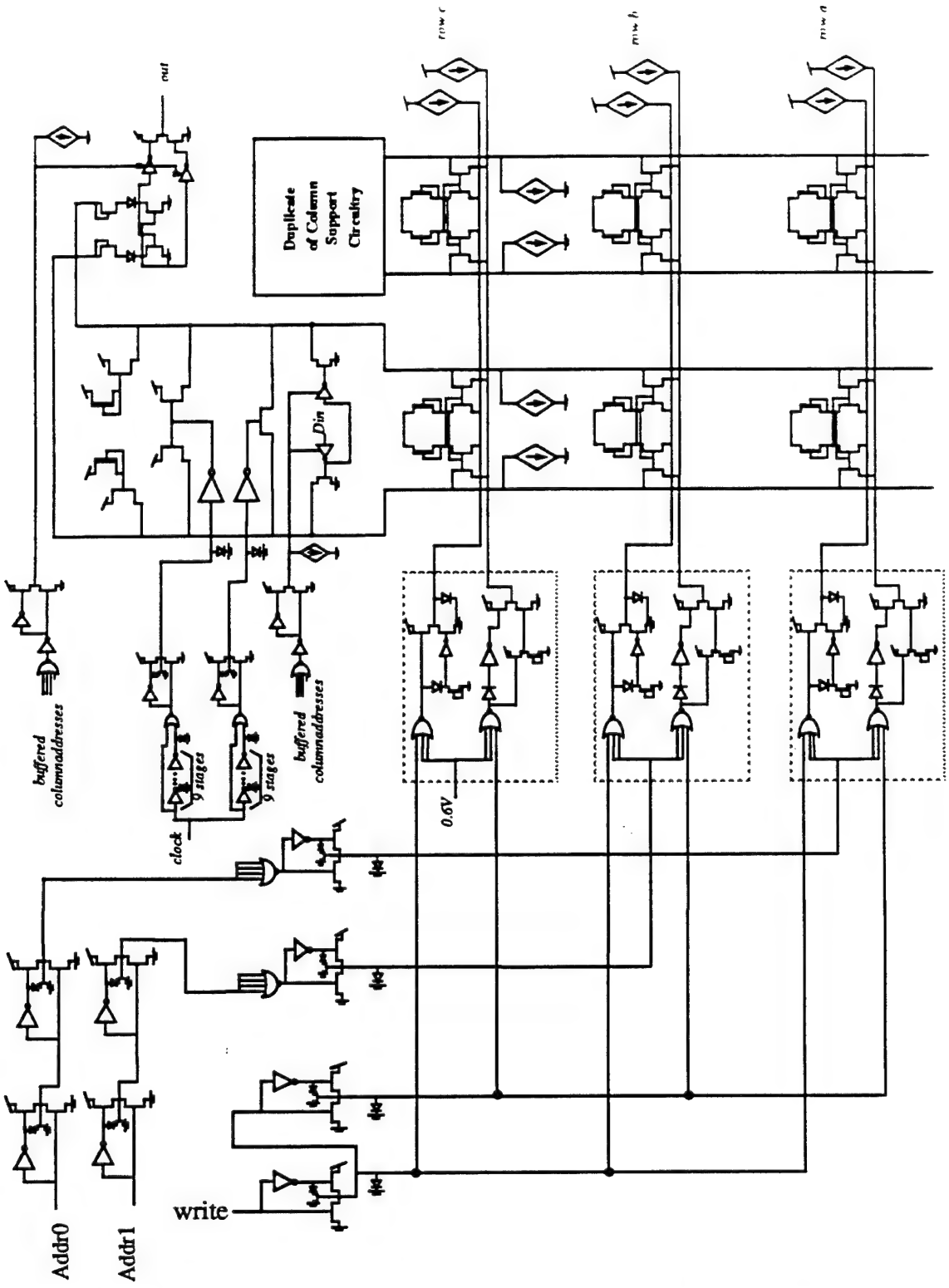


Fig. 5.22: Model of the SRAM with arbitrary column folding.
The model is dynamically adjusted to reduce complexity for the common case of only one column per bit. This involves removing the elements in the shaded box, which are used to model inactive columns.

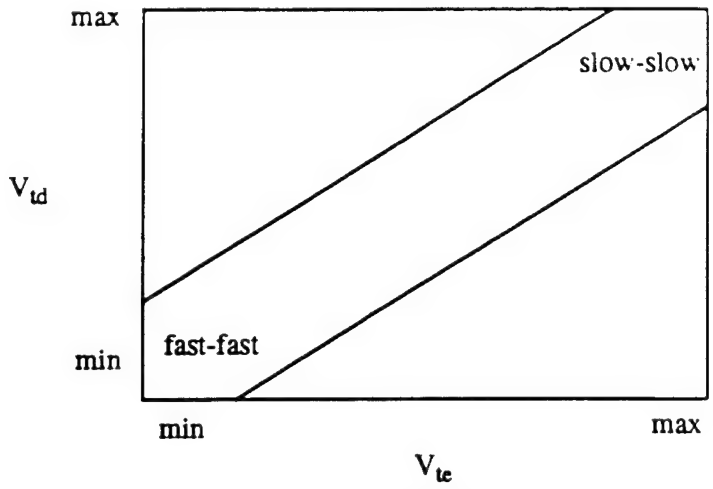


Fig. 5.23: Process space definition using a V_{te}-V_{td} map.

these simulation points are defined by

$$slow\text{-}slow\text{-}corner = (V_{tec} + 3\sigma V_{te}, V_{tdc} + 3\sigma V_{td}) \qquad \text{and}$$

$$fast\text{-}fast\text{-}corner = (V_{tec} - 3\sigma V_{te}, V_{tdc} - 3\sigma V_{td})$$

where V_{tec} and V_{tdc} represent the typical values of enhancement and depletion threshold voltage found in the center of the process space, and σV_{te} and σV_{td} are the standard deviations of the threshold voltages. The three-point model also includes the (V_{tec}, V_{tdc}) point located in the center of the process space.

The six-corner model considers deviation from the longitudinal direction (along the slow-slow to fast-fast line) by also including a perpendicular variation component. The coordinates of the six-point process space include the fast-fast and slow-slow corners used in the two-point model

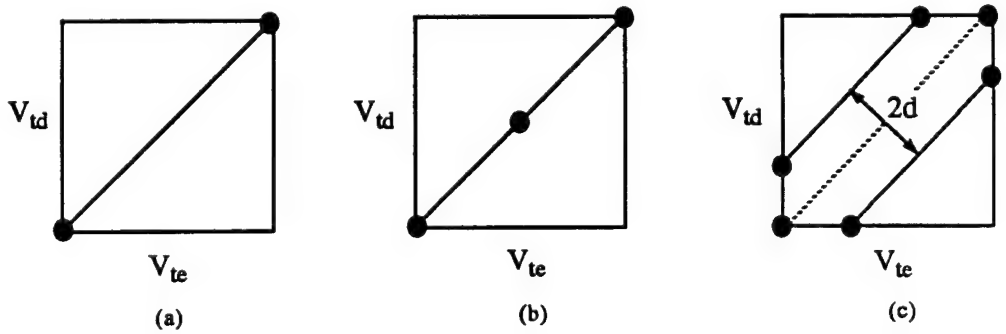


Fig. 5.24: Process space models used for simulation:
(a) two-corner model; (b) three-corner model; (c) six-corner model.

as well as the points

$$(V_{tec} + 3\sigma V_{te} - d\sqrt{2},\ V_{tdc} + 3\sigma V_{td}),$$

$$(V_{tec} + 3\sigma V_{te},\ V_{tdc} + 3\sigma V_{td} - d\sqrt{2}),$$

$$(V_{tec} - 3\sigma V_{te} + d\sqrt{2},\ V_{tdc} - 3\sigma V_{td}),\ \text{and}$$

$$(V_{tec} - 3\sigma V_{te},\ V_{tdc} - 3\sigma V_{td} + d\sqrt{2})$$

where d is the maximum perpendicular direction threshold deviation from the center line.

For most of the simulations, the two-corner model is sufficient for ensuring functionality. The three-corner model is used to obtain additional characterization information. The six-corner model is used for final sizing of the sense amplifier, which is the most sensitive circuit to process variations. This model is also used for final SRAM characterization.

5.7.3 Circuit Simulation

In Section 5.6, several constraints were defined to ensure a process tolerant, functional memory design. In this section, a simulation that has been designed to verify these constraints while using a small number of simulation cycles is described.

Due to the symmetrical nature of the memory cell and the read and write operations, some pairs of constraints can be collapsed into individual constraints. Further, a number of the constraints can be checked simultaneously. For simplicity, the simulation is described assuming a column folding of one. The simulation for a higher amount of column folding is identical, except that cells in the extra columns are checked to ensure that data in nonselected columns is not disturbed.

The simulation consists of exercising two active rows, rows a and b, with one dummy row and associated current mirrors to model the data-dependent bit-line to storage node coupling effect described in section 5.7.1.4. This model was shown in Fig. 5.23.

Fig. 5.26 shows a timing diagram of the simulation. The sequence of operations consists of two writes followed by three reads followed by two writes and one read.

The first two writes check that the same and the opposite type of data can be written to a cell. These writes also check the ability of the memory to write with and against the grain of all of

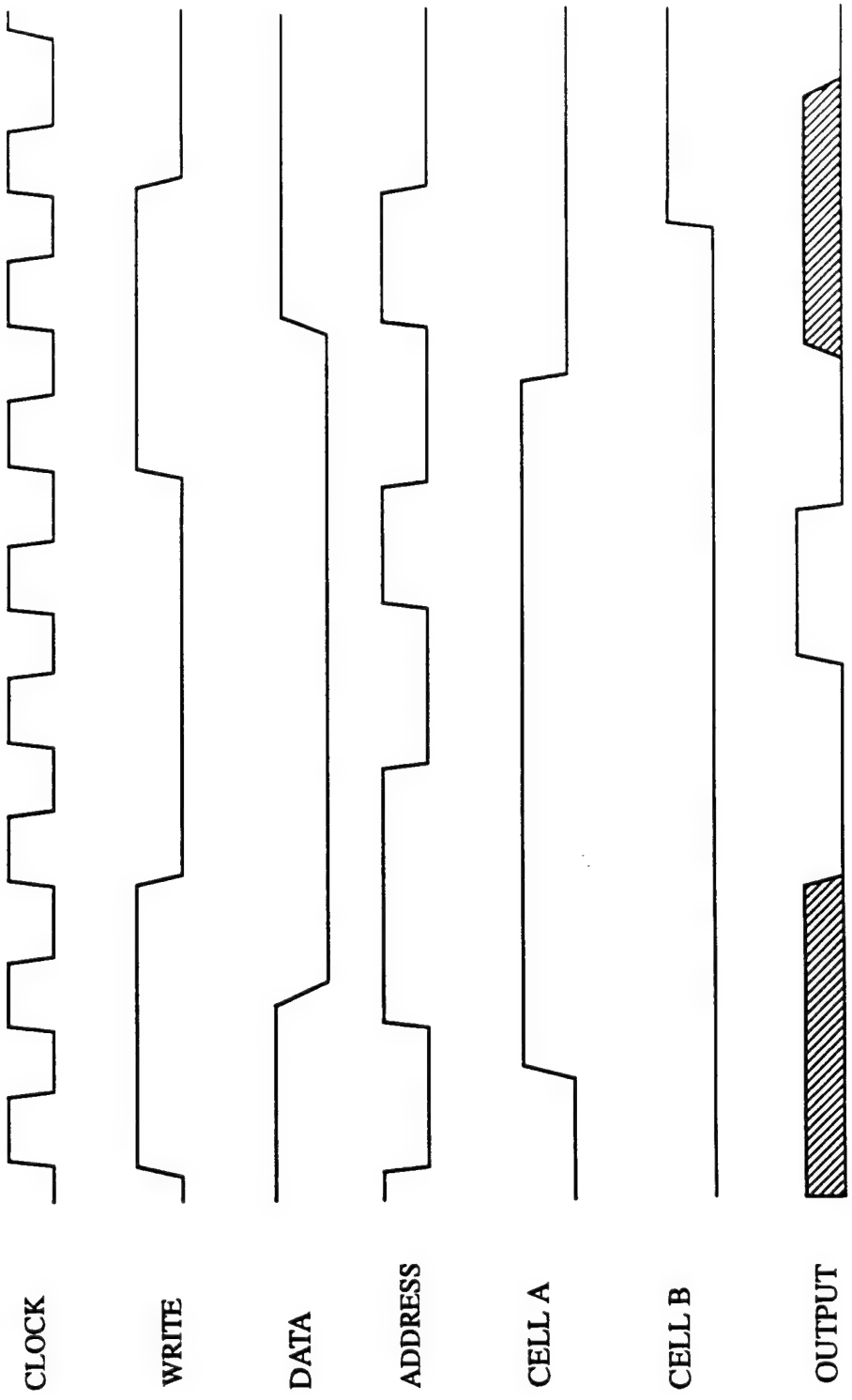


Fig. 5.25: Timing diagram of the simulation used to verify the functionality constraints.

the cells in the column since the bit-line clamping is modeled by the dummy cell and the associated dependent current sources. Due to the symmetry of writing a one and a zero, this first set of two writes actually checks constraints C-1, C-2, C-3, C-4, C-11, C-13, and C-14.

The first read is to the address that was just written. This tests constraints C-7 and C-8. This also checks the ability of the sense-amplifier to read a zero.

The set of three reads performs a read-0, read-1, read-0. This checks constraints number C-9 and C-10.

The next two writes cause new data to be written to each of the memory cells. Thus, the four write operations verify the ability of the memory to write data with and against the grain of data stored in a column. They also verify the ability to write the same and opposite values as previously stored in the cell. Thus, four write operations verify constraints C-1 through C-4 and C11-C14.

The last read is a read-after-write of the opposite type of data within the column. Using symmetry, this checks for constraints C-5 and C-6.

In addition to checking for functionality constraints, this simulation is used to determine the worst-case read and write times. By subjecting the memory to a range of operating sequences, the read access time can be measured after a write of the same type of data, a write of the opposite type of data, and after a read of the opposite type of data. The write times can similarly be measured with and against the grain of a column, and immediately after a read or a write. The resulting values are sorted to identify the worst-case values of these delays across operating sequences. This characterization methodology is much more comprehensive than the methodologies currently employed by memory compilers that use a single "worst-case" path for measuring delays. While it can be argued that a particular sequence of operations produces the "worst-case" delay, the different sensitivities of circuits to process variations make the validity of these arguments suspect across process variations.

5.8 RAM Compiler Transistor Sizing and Characterization Algorithm

In the past few sections, various circuit models for efficient and accurate simulation of the RAM were described; the constraints necessary to produce a functional memory were defined, and

Table 5.1: Parameters used in the auto-transistor sizing and characterization program

Fixed During Initialization	Variable: Affecting READ Only	Variable: Affecting WRITE Only	Variable: Affecting READ and WRITE	Variable: Simulation Parameters
# of rows	equalization cct. precharge inverter size	cell-gnd. driver pullup	cell-gnd. driver pulldown	data hold time
# of columns	equalization cct. pull-together inverter size		write cctry pull-down	address setup time
# of folds	word-line driver pulldown		read/write buffer size	equalization pulse width
memory cell pullup length	sense-amplifier pulldown		predecoder buffer size	
memory cell pullup width	word-line driver pullup		address buffer size	
			pulse generator inverter β	
			pulse generator diode load	

a simulation that exercises a model of the SRAM to verify these constraints was described.

In this section, an algorithm that has been developed to traverse the transistor size space in search of a process tolerant memory that achieves a desired performance goal is described. The first two stages of the algorithm are described by pseudo-code in Fig. 5.27.

5.8.1 Initialization Stage

The first portion of the auto-transistor sizing and characterization program initializes a set of parameters. These parameters are listed in Table 5.1. The parameters that are not varied by the program are listed in the first column.

The second, third, and fourth columns of parameters are transistor sizes that can be varied by the program. These parameters have some effect on the read access time, write time, or both read and write times. Most of the parameters which affect both the read and write times have the same effect on both the read and write times when sized in a given direction. Thus, the shape of

```
/* Stage I: Initialization Procedures */
Parameter_Initialization();
Generate_Necessary_Layout();
Extract_Line_Capacitances();
Determine_Cell_Areas();
Extract_Line_Resistances();
Initialize_Pulsegen_Model();
for (range of sense-amp sizes) {
        Find_Best_Circuit();
}
for (range of pulse-widths) {
        Find_Best_Circuit();
}

/* Stage II: Timing Driven Transistor Size Search */
while ( (taccess > target_access_time) || (twrite > target_write_time) ) {
    /* Check for slow edge rates */
    /* Buffered Address Signals */
    if (edge_rate(buffered_address) > max_edge) {
        ReSize(address_buffer);
    }
    /* Predecode Signal Edge Rates */
    if (edge_rate(predecode_lines)> max_edge) {
        ReSize(predecode_buffer);
    }
    /* Read/Write Signal Edge Rates */
    if (edge_rate(read_write_signals) > max_edge) {
        Resize(read_write_buffer);
    }
    if ((write_time - target_write_time) > (read_time - target_access_time)) {
        /* write time more critical */
        Poll_Write_Circuits();
    }
    else {
        /* read time more critical */
        Poll_Read_Circuits();
    }

    if (no.improvement.possible) {
        exit_while_loop;        }
    Optimize_Senseamp();
    Optimize_PulseWidth();
}
```

Fig. 5.26: Pseudo-code for the initialization stage and timing driven transistor sizing search stage of the transistor sizing and characterization module

the n-dimensional solution space along these axes represents a surface which is well understood. In contrast, an increase in the memory cell access transistor width leads to an improvement in access time, whereas a reduction in memory cell access transistor width leads to an improvement in write time. This opposite effect complicates the transistor size search. The fifth column of parameters listed in the table are timing-related simulation parameters. This set of parameters can be varied by the program.

Once the simulation parameters have been initialized, the next step is to furnish the simulation files with the necessary parasitics. The transistor sizing module first makes calls to the physical design module to generate the necessary layout using the initial transistors sizes, and to extract the capacitances needed for annotating the simulation files.

The physical design module is then called again to provide information on the physical dimensions of the necessary cells based upon the initial transistor sizes. This information is used to extract parasitic resistances needed by the simulation files to model voltage drops and RC delays.

The purpose of the pulse generator is to generate a pulse that is wide enough to satisfy the requirements of the equalization circuitry. A behavioral model of a pulse requires much less simulation time than an explicit representation of the pulse generator circuitry. Thus, during the initialization sequence, the pulse generator is initialized as a behavioral element. During the post-processing stage, this behavioral model is replaced with the actual circuit model to determine the necessary pulse generator parameters.

The last two steps of the initialization stage are a determination of the sense-amplifier size and the pulse generator pulse width that will provide the transistor sizing routine with a process tolerant, functional starting point. In the next section, a parameter selection procedure that is called to perform this transistor size and pulse width search is described.

5.8.2 Parameter Selection Procedure

Decisions need to be made about the best transistor size for a particular circuit, or which one of a number of circuits to size while searching through the n-dimensional transistor size space. The parameter selection is performed by a collection of programs that have been written to con-

struct circuit simulations, to perform the simulations, to analyze the results of the simulations, and to choose the best of the simulation results according to predefined selection criteria.

There are two modes in which the parameter selection procedure is used. In the first mode, the compiler searches for a transistor size (or parameter value), from a range of transistor sizes (or parameter values), that will result in the best access time or write time. In the initialization stage, for example, the compiler searches for a sense-amplifier pulldown transistor size, from a range of possible sizes, that results in the best read and write times.

In the second mode, the compiler has a number of different circuits that can each offer improvements in access time or write time by performing an incremental change in a transistor size. Here, the compiler will choose the circuit that provides the best increase in performance for the lowest cost, subject to a predefined cost criterion function.

In both of these modes, there is a large amount of inherent parallelism because the simulations are independent of each other. To exploit the parallelism, a framework has been developed which first finds the least loaded work stations on a network. Parallel simulations are then distributed to these machines to minimize the time required to compare the relative merits of each circuit perturbation. Once all of the simulations on the remote machines have been completed, the results can be analyzed. The simulation used to exercise the SRAM model was presented in Section 5.7.2. This simulation allows the checking of all of the constraints, defined in section 5.6, that are necessary to produce a functional RAM. After the simulation has been performed, an analysis program checks the results of the simulations to ensure that all of the constraints are met across all process corners in the process model.

There are a number of different ways to determine which one of the simulations gave the best results. One method is to choose the simulation that produced the fastest RAM, without regard to area or power costs. This approach, called a greedy descent, has been incorporated into the compiler as a selection option. A second approach is to choose the simulation which resulted in a speed improvement with the lowest power-delay product of the choices, corresponding to objective function (O-1). Similarly, the delay-area criterion function (O-2) can be used to guide the transistor sizing.

A set of transistor sizes is only considered if all 16 constraints, C1-C16, were observed in

the SRAM simulation using those sizes. Thus, this procedure of selecting transistor sizes or parameter values is guaranteed to produce a process tolerant, functional RAM.

5.8.3 Timing-Driven Transistor Size Search

The initialization produces a simulation model with accurately extracted resistances and capacitances, and a set of transistor sizes that result in a functional design. The second stage is the timing driven transistor size search. In this stage, the transistor size space is methodically searched to meet the desired read access and write times.

The buffers are set to their minimum sizes during the initialization stage. The edge rates of the buffered address lines, the read/write lines, and the predecode lines are first checked in the timing-driven transistor sizing loop. If the rising or falling edge rate for a given line is too large, then the size of the buffer driving that line is increased. The edge rate of a signal can have an impact as great as a 40% on the delay of subsequent stages [Kay93]. Consequently, slow edge rates early in the readout or the write paths can significantly degrade the performance of the memory. The specification of a minimum edge rate on these lines is also a sound approach to minimizing the delay of the first few stages of logic.

The second step is to determine whether the read access time or the write time is more critical. Depending upon which time is more critical, a different set of transistor sizes is considered. If the read access time is more critical, it can be reduced by increasing the word-line driver pull-down transistor size, by increasing either one of the equalization circuitry inverter sizes, by decreasing the cell-ground driver pull-down transistor size, or by increasing the write circuitry pull-down transistor size. Changing transistor sizes inside the memory cell can also reduce the read access time. To minimize computation time, variations in these transistors are not considered until the program is outside of the loop of Fig. 5.27 since such changes require re-extraction of parasitic capacitances and resistances for every perturbation in the transistor sizes.

The impact of independently sizing any one of these parameters is simulated and analyzed using the parameter selection procedure. Based upon the user-selected criterion function (either the fastest solution, the solution with the smallest power-delay product, or the smallest area-delay product), one of the five parameters is chosen for improving the read-access time.

```
/* Stage III: Post-Processing */
Extract_Line_Capacitances();
Determine_Cell_Areas();
Extract_Line_Resistances();
Replace_Pulse_Generator();
Optimize_Pulse_Beta();
Optimize_Pulse_Diode();
Coarse_Optimize_Senseamp();
Fine_Optimize_Senseamp();
if ( (taccess > target_access_time) || (twrite > target_write_time) {
   Optimize_AccessTransistor();
}
Find_Setup_Time();
Find_Hold_Time();
Report_Information();
```

Fig. 5.27: Pseudo-code for the post processing stage of the transistor sizing and characterization module

If the write time is more critical, it can be reduced by decreasing the cell-ground driver pull-down transistor size, increasing the cell-ground driver pullup transistor size, or increasing the write-cell pull-down transistor size. The parameter selection procedure is then used to determine which of these changes leads to the most cost-effective improvement in write time.

If any of the circuitry associated with the bit lines is altered, the desired sensitivity range of the sense amplifier may be impacted. Hence, the sense-amplifier is re-optimized within the loop. Since an increase in the speed of the path may require a modification in the pulse width to achieve better performance, the pulse width is also re-optimized within the loop.

This loop is exited to perform the post-processing once the target access and write times are achieved, or all possible circuit parameters have been improved, or no improvement could be found for the more critical of the read and write time while maintaining the specified noise margins.

5.8.4 Post-Processing

In the timing-driven transistor sizing loop, no changes are made to the memory cell. Therefore, the parasitics associated with the bit lines, word- and cell-ground lines, and predecoder lines do not change dramatically. In the post-processing stage, described in Fig. 5.28, the first

operation is to re-extract capacitances and line resistances based upon the new transistor sizes found within the loop.

5.8.4.1 Pulse Generator Sizing

The next step after re-extracting the parasitic resistances and capacitances is to replace the behavioral model of the pulse generator with the pulse-generator circuitry. A two-step search is used to determine the pulse-generator transistor sizes. In the first step, the diode loads are fixed in the middle of their range and an optimal β value for the inverter chain in the pulse generator is found. This search for a β that produces the highest memory performance is performed using the parameter selection procedure on a range of discrete β values.

In the second step, the optimal β is used to find a diode load which results in the best memory performance. Again, the parameter selection procedure is used to determine an appropriate diode load.

5.8.4.2 Sense-Amplifier Sizing

Using two- or three-corner process models in the steps described above, highly process tolerant circuits are achieved. The most sensitive circuit to process variations is the sense-amplifier. Therefore, in the post-processing stage, the process model is expanded from the two- or three-corner model to the six-corner model to achieve a more robust sense-amplifier sizing. Since the parameter selection procedure can perform many simulations in parallel, a divide-and-conquer search is used to quickly find the optimal sense-amplifier transistor size.

5.8.4.3 Access Transistor Sizing

The read-access time is reduced when the access transistor size is increased, and the write time is reduced when the access transistor size is decreased. If the target read or write time is still not met after performing the pulse-generator sizing and the sense-amplifier sizing, the impact of modifying the access transistor size is examined. Using the parallelism of the parameter selection procedure, a divide-and-conquer search is performed to quickly find the access transistor size that will come closest to meeting the desired goals. A slight complication is introduced by changing

the memory cell size. Changing any size in the cell has an impact on the parasitics on most of the global signal lines. Thus, for each access transistor size considered, all necessary capacitances and resistances needed for simulation are re-extracted.

5.8.4.4 Memory Characterization

The final step in the automatic transistor sizing and characterization module is to characterize the memory. This characterization includes determining the necessary setup and hold times in the RAM. The address setup and data hold times, listed in Table 5.1 on page 150, are used in the behavioral pulse generators which exercise the address and data ports of the RAM.

As long as the address for a write is presented a setup-time before the write signal, data from the previous address location will not be inadvertently written. The setup time is actually a negative number, meaning that the address can be presented after the write signal is asserted without risking a write to the previous address. To determine this setup time, a divide-and-conquer search is performed over a range of address setup times to find the minimum value at which the memory fails.

There is also a hold time requirement for data with respect to the trailing edge of the write signal. If the data changes too quickly after the write signal is removed, then the new data may be written to the previous memory address. This hold time is determined using a divide-and-conquer search to find the minimum hold time that will result in a functional memory.

5.9 Examples of Generated Memories

The RAM compiler was used to generate a 128x77 bit instruction cache for the Aurora III microprocessor. To examine the trade-offs in speed and power dissipation, the compiler was run once using a memory cell pullup transistor length of 12µm, and once using a pullup length of 5.6µm. These sizes are the boundary values allowed by the compiler, and hence also represent the boundaries of performance that the SRAM can attain. An 85mV control of threshold voltage was assumed for these simulations.

Fig. 5.29 shows the power-delay points that were achieved by the compiler for memories made with the two lengths of memory cell pullup transistor. In this figure, the average operating

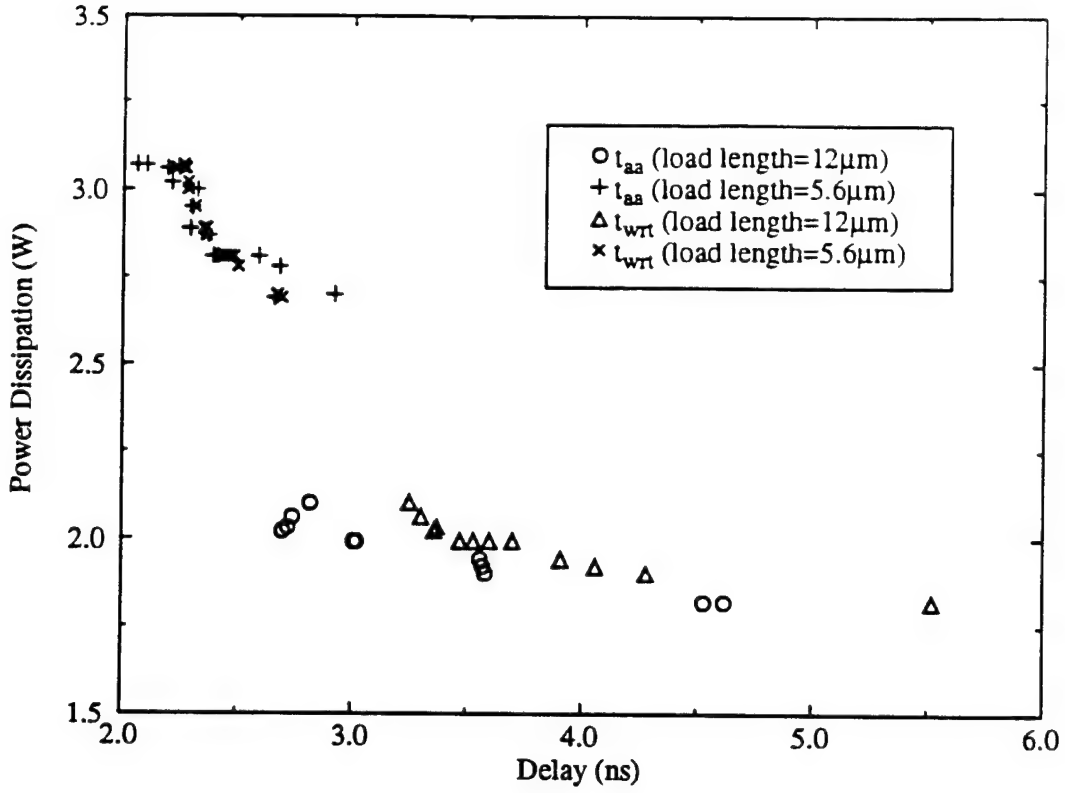


Fig. 5.28: Power-delay characteristics for a 128x77 SRAM using minimum-sized and maximum-sized load transistor pull-up lengths.

power dissipation is plotted against the slowest read access time (t_{aa}) and write time (t_{wrt}) achieved over the constraint-driven simulation and across process variations. Using minimum sized gates, the memory optimization starts at its lowest-power state. The optimization proceeds from right to left on the power-delay graph because the speed and power dissipation increase with the number of iterations.

The strength of the memory cell pull-up device significantly impacts the performance of the RAM. The 5.6µm long device can achieve a read or a write in 2.3ns for a 434MHz operation. Increasing the length of this pull-up device to 12µm only guarantees a read or a write operation in 3.3ns for a 300MHz operation. SRAM area is plotted against cycle time using these load transistor lengths in Fig. 5.30. This graph shows that the total area is not significantly affected during transistor sizing. The figure also illustrates that the memory cell pullup device dictates the total area of these SRAMs. The SRAM using the 12µm pull-up device requires 7.8 mm^2, compared to only

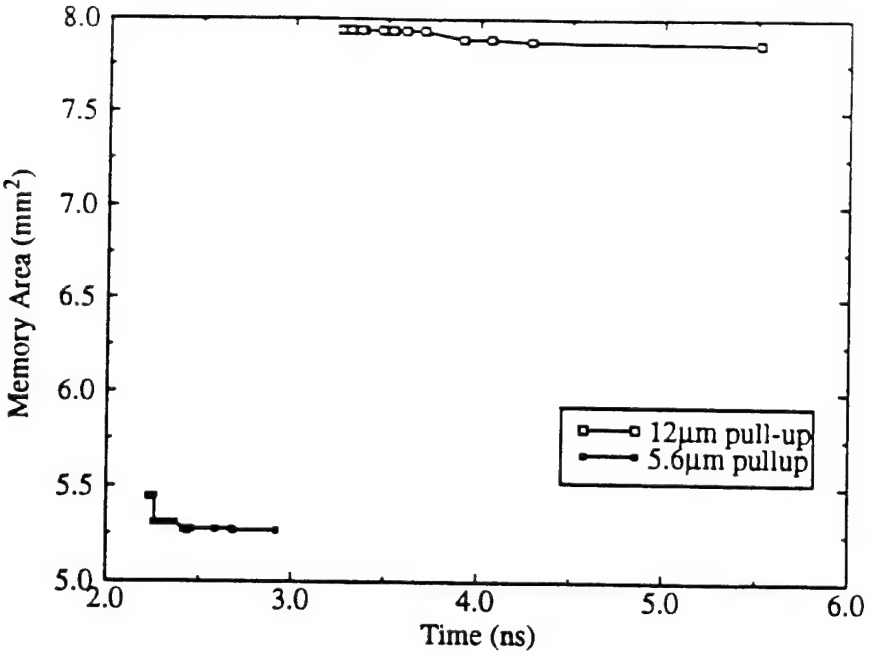


Fig. 5.29: Memory area versus cycle time during transistor sizing.

5.45mm^2 for the SRAM using the 5.6µm long pull-up device. The smaller area, however, is achieved at the expense of greater power dissipation. Fig. 5.29 shows that reducing the entire area of the SRAM by 32% and reducing the access time by 30% requires a 47% increase in power dissipation.

The components of SRAM delay, power dissipation, and area for the SRAM compiled using a 5.6µm pull-up device memory cell are given in Figs. 5.31(a), (b), and (c). These graphs show how power and area were traded to achieve smaller read and write times. Each component in these figures is graphed cumulatively; the curves at the tops of these graphs show the total power, delay, and area.

Fig. 5.31(a) shows that the major component of power in this SRAM is dissipated in the memory cell array. The higher performance of these memories over the 12mm-pullup device memories is partly due to the smaller cell area, and partly due to the dependence of the memory cell access transistor characteristics on cell current. In the next section, the relative contributions of these components will be examined.

The second largest component of power dissipation is in the word drivers. During write operations, bit-line to word-line coupling through the memory cell access transistors draws current

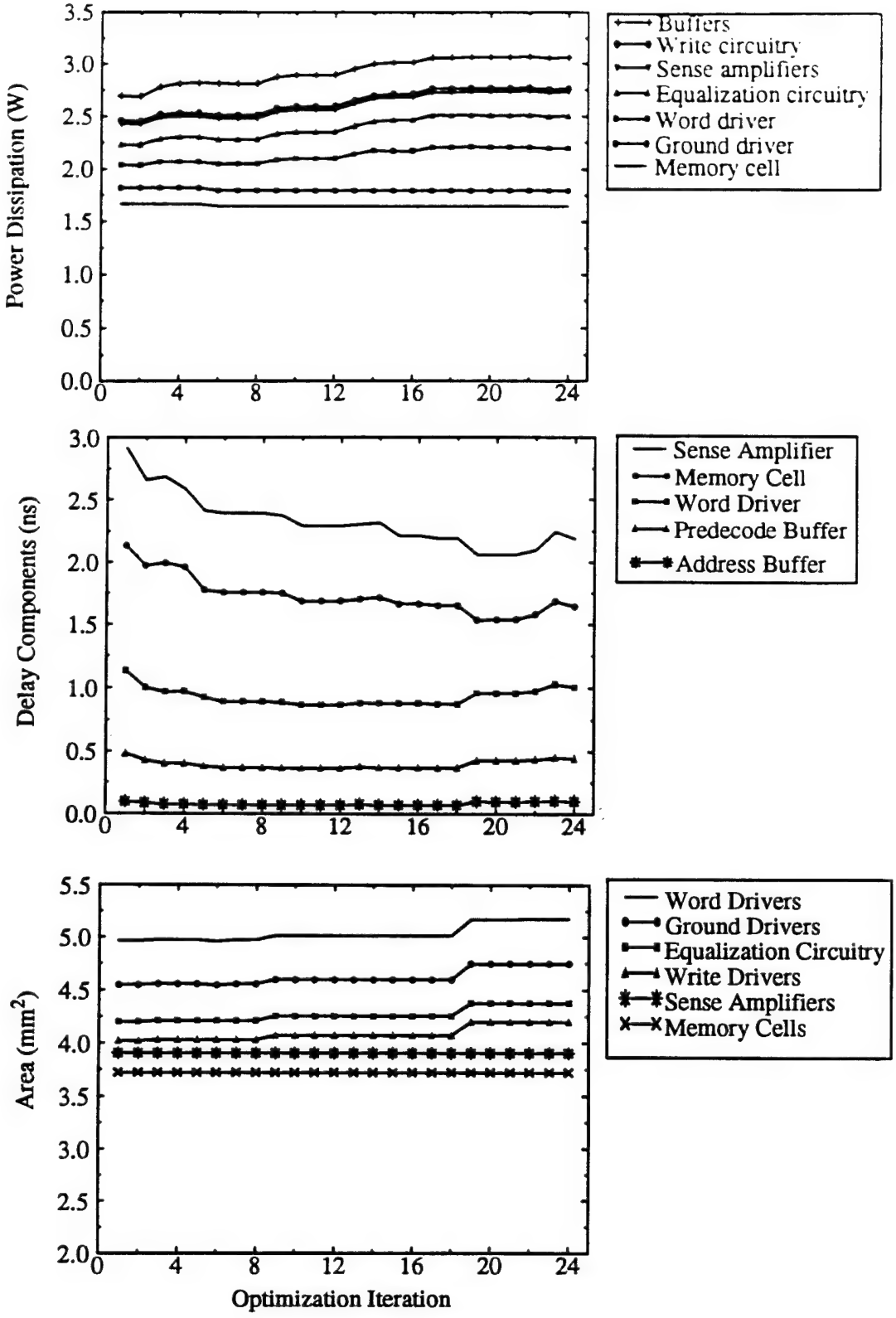


Fig. 5.30: Components of delay, power dissipation, and area for the 128x77 cache SRAM.

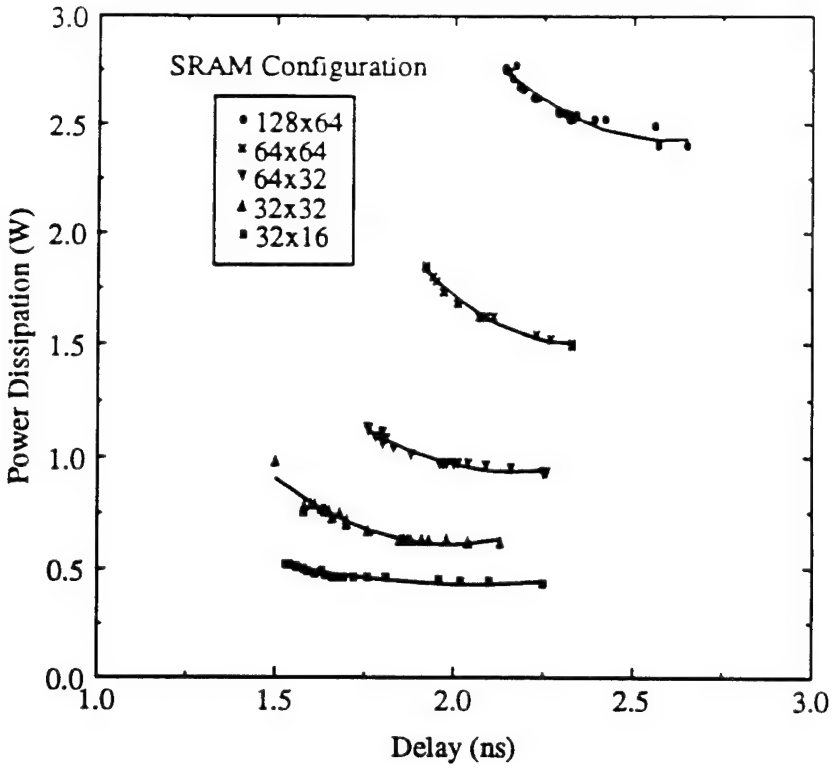


Fig. 5.31: Compiler generated power-delay trade-off curves.

from the word drivers to maintain the word line-high voltages. This higher power dissipation component is, therefore, unavoidable with the CMMC.

The delay chart shows a monotonically decreasing delay during the inner loop of the timing-driven transistor sizing. The post-processing occurs above iteration number 19. During post-processing, the read access time delay increases due to an expansion of the process model, and an effort to trade the write time for the read time within the memory cell.

Fig. 5.31(c) shows the area components of circuits in the main SRAM block. The cell area accounts for 72% of the total SRAM area. The next largest components of area are the word and cell-ground drivers. This is partly due to the many predecode lines that run through the cell, and partly due to the large transistors which pull the cell-ground and word- lines low.

Several memory configurations with different numbers of rows and columns were generated by the compiler. The power-delay curves for several SRAMs that were generated are shown in Fig. 5.32. The figure shows that the larger memories exhibit a greater power trade-off for speed. On a per-bit basis, all memories actually exhibit a similar trade power per bit for speed. These

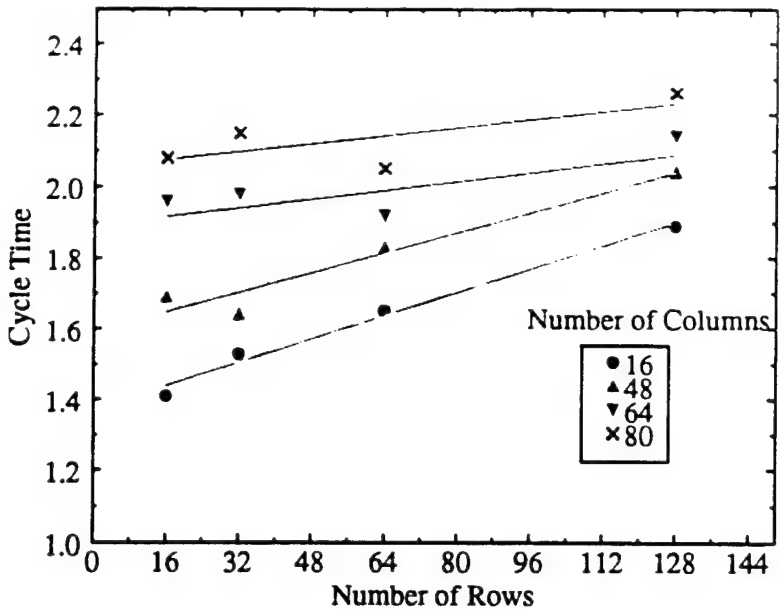


Fig. 5.32: Minimum cycle time found as a function of the number of rows and columns of the SRAM.

curves show that the compiler can offer substantial trade-offs in speed for power. There is no significant increase in delay with column folding because a single sense amplifier is attached to each column.

Fig. 5.33 shows the minimum cycle time achieved as a function of the numbers of rows and columns. For larger memories, the access times are determined by the number of rows more-so than the number of columns. This graph shows that the compiler is capable of generating memories as large as 10kb with 2.25ns access times. Not surprisingly, for smaller memories of a given size, the preferred organizations are those with a larger number of rows and a fewer number of columns, because the higher bit-line capacitances and the more heavily loaded predecode lines are much harder to drive than the word lines and cell-ground lines.

Fig. 5.34 shows layout plots of compiler-generated 16x16 (256b), 64x64 (4kb), and 128x64 (8kb) SRAMs. The 4kb SRAM was generated using manual standard-cell placement, while the other two SRAMs were generated using automatic placement and routing of the standard cells.

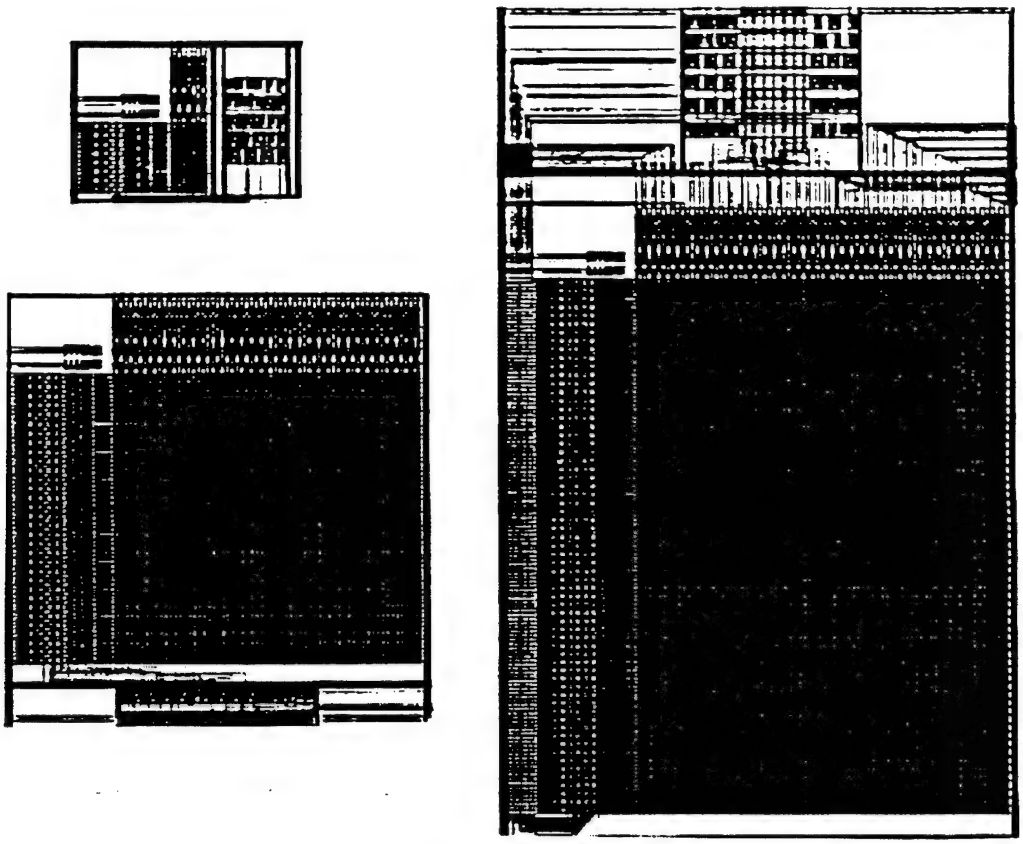


Fig. 5.33: Layout plots of compiler-generated 256b, 4kb and 8kb SRAMs.

5.10 The RAM Compiler as an Analysis Tool

As discussed in Section 5.3, the physical design module and the transistor sizing module form a framework that can readily adapt to changes in process technology. Memories can be optimized in different processes by simply updating the process information files. This feature of the compiler was used to determine the impact of technology trends on the cycle time.

Fig. 5.35 shows the cycle time of an 8kb SRAM as functions of percent reductions in line resistance, intrinsic gate delay, and line capacitance on the cycle time. This figure shows that the performance is dominated by interconnect loading and intrinsic gate delay. A reduction in interconnect capacitance increases circuit speed more effectively than a reduction in intrinsic gate delay.

This graph also shows that RC delays and I-R drops along the word-lines have an appreciable impact on the performance of the SRAM. The push for denser digital circuits has been accomplished partly through a reduction in wire pitch. Extrapolation from this graph indicates that

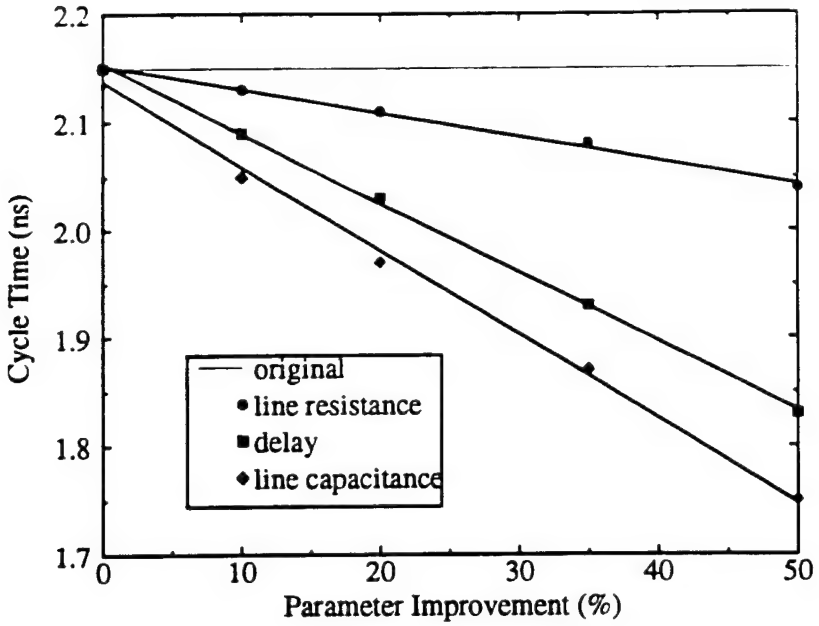


Fig. 5.34: 8kb SRAM cycle time sensitivity to lineresistance, raw gate delay and line capacitance.

at present integration levels, a reduction in line width (which leads to an increase in line resistance) will cause degradation in performance. This figure also shows that reductions in wiring pitch will cause a greater improvement in memory cycle time due to decreased line capacitance than the degradation in memory cycle time due to increased wire resistances. The relative merits of the improvements and degradations in performance must be carefully considered when investing resources to improve circuit densities and performances.

Fig. 3.36 shows the impact of cell area reduction on memory performance. For these simulations, a uniform reduction in cell dimensions was assumed. Changes in line capacitances and resistances reflected only reductions in cell area.

The baseline RAM cell area for these simulations was 400μm^2. An eight-fold reduction in cell area results in cell dimensions that are about $\sqrt{8} = 2.8$ times smaller. Since both the parallel-plate and fringe components are simultaneously reduced. This graph shows that a 30% reduction in cycle time or a 43% increase in clock rate can be achieved when cell areas reach levels that are comparable to those currently enjoyed by CMOS. This graph also illustrates that dramatic increases in memory speed can be achieved by focusing processing efforts on reducing line resistances and capacitances alone, without regard to intrinsic gate delay.

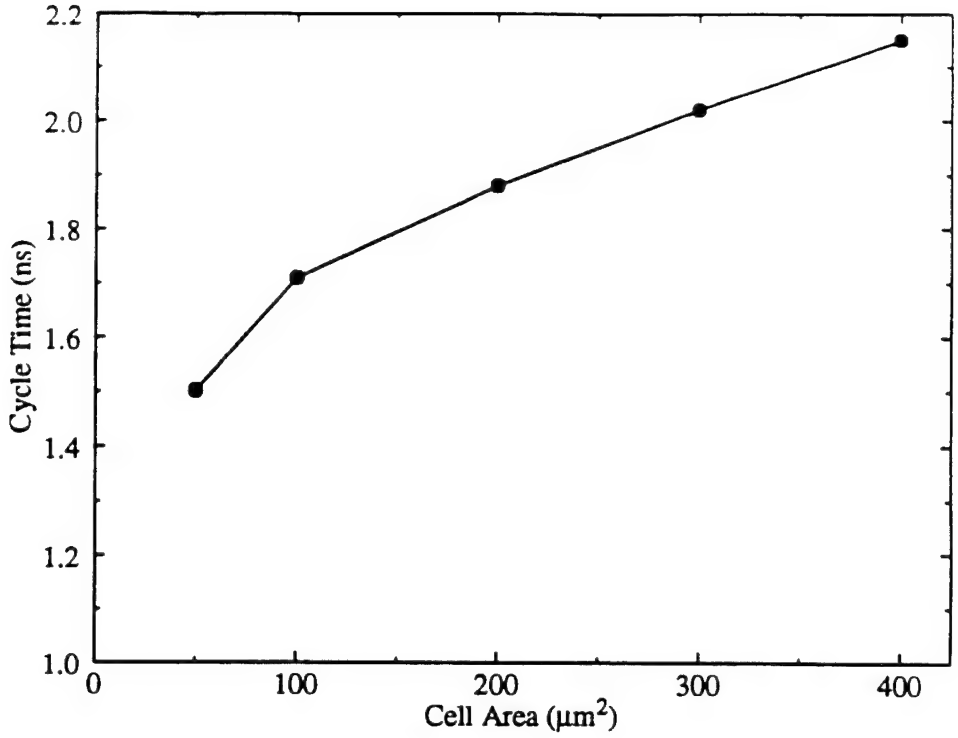


Fig. 5.35: 8kb SRAM cycle time sensitivity to cell area.

5.11 Conclusions

In this chapter, the Aurora RAM Compiler (ARC) was described. This compiler was developed to generate and characterize highly manufacturable optimized SRAMs using GaAs E/D MESFET technology. ARC represents a departure from the conventional methodologies used in RAM compilers. Because of the low noise margins and large process variations in GaAs circuits, this compiler uses HSPICE as a simulation engine for performing delay as well as signal noise margin calculations.

By using a circuit simulator to perform delay and signal calculations, the cost of developing macromodels, the cost of maintaining a transistor-level simulator within the compiler, and the cost of process-tolerant circuit evaluation were eliminated. The costs associated with this scheme are the development of accurate transistor models and the time required for computation. Approximately 5 hours were needed to size, characterize, and generate a process tolerant SRAM on a small network of HP 9000/710 machines.

Several modeling techniques were developed to efficiently predict effects that can degrade

the performance of memories using the CMMC. A process tolerant design flow has been developed for this RAM compiler which performs simulations over a sequence of operations and process corners to achieve a high design yield. As transistor nonuniformities become problematic with feature sizes below 0.35μm, CMOS compilers will also benefit from this new approach.

Simulations indicate that the compiler can produce memories as large as 10kb with 2.25ns access times. This is significantly faster than what has been achieved for stand-alone or embedded E/D MESFET GaAs SRAMs to date. An analysis of the impact of process improvements on SRAM performance indicates that reductions in cell capacitance will have the greatest impact on memory speeds when compared to improvements in device transconductance, and line resistances.

CHAPTER VI
CONCLUSIONS

Advances in circuit design and processing technologies have driven SRAM performance in technologies such as CMOS and BiCMOS. To leverage the benefits of a technology, it is necessary to develop design principles and circuit structures that are best suited for that technology. The focus of this thesis has been to develop new circuit structures and design methodologies to produce higher performance, lower power, process tolerant SRAMs and digital circuits using E/D MESFETs in GaAs. This work has led to several significant contributions.

6.1 Contributions

Some of the problems that have plagued GaAs E/D MESFET memories are large memory cell sizes, high subthreshold leakage currents, and destructive readout. In this thesis, the current mirror memory cell (CMMC) has been presented as a solution to many of these problems. The CMMC occupies a smaller area than a conventional 6-transistor memory cell. Faster read and write operations are achieved for memories designed with the CMMC than for memories designed with a conventional memory cell. The biasing scheme for this cell achieves the maximum suppression of leakage currents associated with access transistors in nonselected rows. These advantages are attained while using only a single supply voltage. Two experimental SRAMs that use the CMMC were successfully designed, fabricated and tested. These memories proved the viability of this approach.

Based on the test results for these two SRAMs and on extensive simulations, several failure mechanisms for CMMC SRAMs were identified. These mechanisms, which include leakage currents, bit-line to word-line coupling, resistive drops in the word line, and charge

injection into the memory cell from the word line, can be avoided by careful design. A test procedure was identified that adequately tests for these faults as well as cell-coupling faults and all stuck-at faults.

The design of process-tolerant circuits requires an understanding of circuit failures in the presence of process variations. Parametric factors that lead to failures in DCFL circuits and in super-buffers that use feedback were examined. A method for calculating the parametric yield of circuits in the presence of process variations was formulated. This was applied to determine the threshold voltage control required for DCFL circuits to achieve 100% parametric design yield. This yield calculation methodology can be applied to optimizing target enhancement and depletion transistor thresholds for achieving high circuit yield, to determine the process parameter control needed to achieve desired yield for circuits of given integration levels, and to make the necessary trade-off in selecting β values.

High static power dissipation hinders the integration levels of memories that use GaAs E/D MESFETs. This thesis presents a new logic style, called power rail logic (PRL), that was invented to reduce some of the standby power of inactive circuits.

In this thesis, a methodology was developed for the characterization and comparison of circuits using process tolerance as a central part of the analysis. The small logic swings, low noise margins, and sensitivity to process variations make such an approach essential for the comparison of GaAs circuits. This methodology was used to compare the merits of PRL to DCFL circuits.

PRL offers circuits with a smaller area and up to 40% lower power-delay products than can be achieved with DCFL. The effectiveness of this logic style has been demonstrated for some of the most common digital logic circuit datapath elements, including multiplexors, latches and flip-flops. PRL can also simplify complex random logic gates, as illustrated by the exclusive-OR gate.

A test chip containing 32-bit DCFL and PRL barrel shifters was designed, fabricated, and tested. The PRL circuit, which was about 12% smaller, was found to operate 13% faster than the DCFL circuit while consuming an average of 24% less power, resulting in a 34% smaller power-delay product. This demonstration vehicle proves the viability of PRL.

Finally, this thesis presented the Aurora RAM Compiler (ARC) which was developed to

generate and characterize highly manufacturable optimized SRAMs using GaAs E/D MESFET technology. ARC builds on all the novel circuit structures, design methodologies, and GaAs design experience described in this dissertation. Memories designed in the ARC use the CMMC for high speed and reliable operation. ARC also uses PRL to reduce the power dissipation in the SRAM read and write circuitry. Further, ARC builds on the process tolerant design methodologies that were developed to generate robust sense-amplifier designs, and to determine transistor sizes that optimize the power-delay products.

ARC represents a departure from the conventional methodologies used in RAM compilers. Because of the low noise margins and large process variations in GaAs circuits, this compiler uses HSPICE as a simulation engine for performing delay as well as signal noise-margin calculations. By using a circuit simulator to calculate delays and signal levels, the cost of developing macromodels, the cost of maintaining a transistor-level simulator within the compiler, and the cost of process-tolerant circuit evaluation were eliminated. The costs associated with this scheme are the development of accurate transistor models and the time required for computation.

Several modeling techniques that efficiently predict effects that can degrade the performance of memories designed using the CMMC were presented in this thesis. To take advantage of the circuit simulator, a design flow was developed for this RAM compiler in which simulations are performed over a sequence of operations and process corners to achieve a high design yield. As transistor uniformities become problematic with smaller feature sizes, CMOS compilers will also benefit from this approach.

This compiler was used to examine the impacts of process improvements on circuit performance. Reduction of line capacitances will have a greater impact on total SRAM performance than improvements in device speeds. Simulation results show that the contribution of line resistance will play an increasing role in limiting memory performance with reductions in wire pitch.

6.2 Future Work

The CMMC allows GaAs MESFET SRAMs to achieve speeds that are very competitive with silicon memories. Poor integration density is the major problem with GaAs E/D MESFET

SRAMs. To illustrate this point, the latest DEC Alpha chip spends 6.3 million of its total 9.3 million transistors on SRAM. The largest SRAM densities reported by GaAs vendors is 50kb of embedded SRAM. The market share of GaAs VLSI circuits will not grow beyond niche applications unless the integration levels for SRAM increase significantly.

Although the CMMC is a necessary step to achieve smaller memory cells in GaAs, it is not enough. GaAs foundries need to offer hidden load devices, transistors with smaller source/drain contact areas, and closer transistor spacing than is currently used in all GaAs production lines. Without such advances, the lack of high levels of integration for memory will prevent GaAs technologies from having any impact in the microprocessor market.

A secondary factor that will limit SRAM integration densities is power dissipation. The cell current must be large enough to offset subthreshold leakage currents in the memory cell. Until the leakage currents are brought well below their present levels, they will also limit the integration levels of GaAs MESFET SRAMs.

There is a continual push in the semiconductor industry toward lower supply voltages for reducing chip power dissipation. The major trend in GaAs technologies is to 1V supply voltages. The CMMC was designed for a 2V supply. New circuit architectures will need to be developed that can allow high-speed operation at reduced supply voltages.

The study of DCFL circuit yield showed that very tight threshold voltage control is necessary to achieve 100% parametric yield. All semiconductor technologies are moving towards smaller feature sizes, resulting in less predictable control of parameters such as threshold voltage. The methodology described for parametric yield calculation can be extended to analyze the trade-offs in speed, power dissipation, cell area, and parametric yield when making process enhancements. As other technologies such as CMOS move toward lower supply voltages, low noise margins will become problematic. An extension of this analysis will help determine the processing requirements needed for reliable low-voltage CMOS operation.

A design methodology was presented for characterizing PRL circuits. In this method, a set of transistor sizes was determined that trace an optimized power-delay curve for a given circuit. This method could be used to great advantage in GaAs compilers by embedding transistor sizing rules into delay calculation routines so that delay can intelligently be traded for power both within

cells and between cells in a design. Significant power savings could result in non-critical paths of a design using this technique.

Power rail logic promises a substantial power savings for digital logic circuits. Even if 40% of the logic on a chip can use PRL, it can lead to a 16% reduction in the power-delay product of the chip. This difference can influence the viability of a chip. An extension of this work will be to examine its applications to complex gates such as adder structures. The usefulness of this logic style will lie in the creativity of designers to find its best applications.

In the Aurora RAM Compiler, the transistor sizing problem was cast as a non-linear optimization problem. A systematic approach was developed to traverse the transistor-size solution space. An interesting follow-up to this effort would be to analyze the benefits of applying classical non-linear optimization approaches to this problem.

In conclusion, this thesis has demonstrated that new integrated circuit technologies can benefit from the development of circuit structures and design principles that are best suited for that technology. If the appropriate amount of time is spent in finding solutions to the technological problems that face it, there is no question that GaAs can be a viable technology for VLSI circuits. I hope that the results contained in this thesis will motivate other researchers to continue to find solutions that will help GaAs to achieve its potential.

BIBLIOGRAPHY

[Abi83] M. S. Abir and H. K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," *Computing Surveys of the ACM*, vol. 15, pp. 175-198, 1983.

[Bla91] T. Blalock and R. Jaeger, "A High-Speed Clamped Bit-Line Current-Mode Sense Amplifier," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 542-548, 1991.

[Bla92] T. Blalock and R. Jaeger, "A High-Speed Sensing Scheme for 1T Dynamic RAM's Utilizing the Clamped Bit-Line Sense Amplifier," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 618-625, 1992.

[Cha91] T. Chappell, B. Chappell, and S. Schuster, et. al, "A 2-ns Cycle, 3.8-ns Access 512kb CMOS ECL SRAM with a Fully Pipelined Architecture," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1577-1585, 1991.

[Con88] J. Conger, A. Peczalski, and M. Shur, "Subthreshold Current in GaAs MESFET's," *IEEE Electron Device Letters*, vol. 9, pp. 128-129, 1988.

[Dob92] D. Dobberpuhl, et. al, "A 200-MHz 64-b Dual-Issue CMOS Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1555-1567, 1992.

[End91] K. Endo, T. Matsumura, and J. Yamada, "Pipelined, Time-Sharing Access Technique for an Integrated Multiport Memory," *IEEE Journal of Solid-State Circuits*, vol. 26p, pp. 549-554, 1991.

[Fie86] A. Fiedler, J. Chun, R. Eden, and D. Kang, "A GaAs 256x4 static self-timed random access memory," presented at GaAs IC Symposium, 1986.

[Fie88] A. Fiedler, J. Chun, and D. Kang, "A 3ns 1Kx4 Static Self-Timed GaAs RAM," presented at GaAs IC Symposium, 1988.

[Fie90] A. Fiedler and D. Kang, "A GaAs Pin-for-Pin Compatible Replacement for the ECL 100474 4K SRAM," presented at IEEE GaAs IC Symposium, 1990.

[Fla90] S. Flannagan, P. Pelley, and N. Herr, et. al, "80ns CMOS 64Kx4 and 256Kx1 SRAM's," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1049-1056, 1990.

[Ful91] D. Fulkerson, "Feedback FET Logic: A Robust, High-Speed, Low-Power GaAs Logic Family," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 70-74, 1991.

[Gab87] B. Gabillard, et. al, "A 1K GaAs SRAM with 2ns cycle time," presented at ISSCC Technical Digest, 1987.

[Gwe93] L. Gwennap, "SGI Provides Overview of TFP CPU," *Microprocessor Report*, vol. 7, pp. 12-13, 1993.

[Hay84] T. Hayashi, et al., "ECL compatible GaAs SRAM circuit technology for high performance computer application," presented at GaAs IC Symposium, 1984.

[Hay85] T. Hayashi, H. Tanaka, H. Yamashita, N. Masuda, T. Doi, J. Shigeta, N. Kotera, A. Masaki, and N. Hashimoto, "Small access time scattering GaAs SRAM technology using bootstrap circuits," presented at Digest of 1985 GaAs IC Symposium, 1985.

[Hei90] W. Heimsch, R. Krebs, B. Pfaffel, and K. Ziemann, "A 3.8-ns 16K BiCMOS SRAM," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 48-54, 1990.

[Hin91] R. S. Hinds, S. R. Canaga, G. M. Lee, and A. Choudhry, "A 20K GaAs Array with 10K of Embedded SRAM," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 245-256, 1991.

[Hir84] M. Hirayama, M. Ino, Y. Matsuoka, and M. Suzuki, "A GaAs 4Kb SRAM with direct coupled FET logic," presented at ISSCC Digest of Technical Papers, 1984.

[Hir86] M. Hirayama, et al., "A GaAs 16Kbit static RAM using dislocations free crystal," *IEEE Transactions on Electronic Devices*, vol. ED-33, pp. 104-110, 1986.

[Hod83] D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits*, 1st ed. New York: McGraw-Hill, 1983.

[Hoe91] D. Hoe and C. A. T. Salama, "Dynamic GaAs Capacitively Coupled Domino Logic (CCDL)," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 844-849, 1991.

[Huf94] T. Huff, "A Latency Tolerant GaAs Floating Point Processor," PhD. Dissertation, University of Michigan, 1995.

[Iiz90] T. Iizuka, "Embedded Memory: A Key to High Performance System VLSIs," presented at 1990 Symposium on VLSI Circuits, 1990.

[Ino82] M. Ino, M. Hirayama, K. Ohwada, and K. Kurumada, "GaAs 1Kb static RAM with E/D MESFET DCFL," presented at GaAs IC Symposium, 1982.

[Ish90] K. Ishibashi, T. Yamanaka, and K. Shimohigashi, "An alpha-Immune, 2-V Supply Voltage SRAM Using a Polysilicon PMOS Load Cell," *IEEE Journal of Solid State Circuits*, vol. 25, pp. 55-60, 1990.

[Kat85] N. Kato, M. Hirayama, K. Asai, Y. Matsuoka, K. Yamasaki, and T. Ogino, "A high density GaAs static RAM process using MASFET," presented at IEDEM, 1985.

[Kay93] A. Kayssi, "A Methodology for the Construction of Accurate Timing Macromodels for Digital Circuits," : Michigan, 1993.

[Kur91] I. Kurowsawa, H. Nakagawa, M. Aoyagi, S. Kosaka, and S. Takada, "A Fully Operational 1-kb Variable Threshold Josephson RAM," *IEEE Journal of Solid State Circuits*, vol. 26, pp. 572-577, 1991.

[Lar90] W. Larkins, "Group III-V Semiconductor Devices with Improved Switching Speeds," Vitesse Semiconductor Corporation, U.S. Patent 4,935,647, June 19, 1990.

[Las93] S. Lassen, S. Long, and K. Nary, "Ultralow-Power GaAs MESFET MSI Circuits Using Two-Phase Dynamic FET Logic," *IEEE Journal of Solid State Circuits*, pp. 1038-1045, 1993.

[Lee93] F. Lee, Personal Communications, 1993.

[Leh84] K. Lehovec, R. Zuleeg, and J. K. Notthoff, "Charge effects in GaAs semi-insulating substrates due to pulsed ionizing radiation," presented at GaAs IC Symposium, 1984.

[Lon89] S. Long and S. Butner, *Gallium Arsenide Digital Integrated Circuit Design*, 1st ed. New York: McGraw-Hill Publishing Company, 1989.

[Mak90] H. Makino, et. al, "A 7ns/850mW 4Kb SRAM Fully Operative at 75°C," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1232-1238, 1990.

[Mat89] S. Matsue, H. Makino, and N. Minoru, et. al, "A Soft Error Improved 7ns/2.1W GaAs 16Kb SRAM," presented at GaAs IC Symposium, 1989.

[Mit84] A. Mitonneau, M. Rocchi, I. Talmud, J. C. Mauduit, and M. Henry, "Direct experimental comparison of submicron GaAs and Si NMOS MSI digital IC's," presented at GaAs IC Symposium, 1984.

[Miy84] H. Miyanaga, S. Konaka, Y. Yamamoto, and T. Sakai, "A 0.85ns 1Kb bipolar ECL RAM," presented at Extended Abstracts 16th Int. Conf. on Solid State Devices and Materials, 1984.

[Miz84] T. Mizoguchi, N. Toyoda, K. Kanazawa, Y. Ikawa, T. Terada, M. Mochizuki, and A. Hojo, "A GaAs 4K bit static RAM with normally-on and off combination circuit," presented at GaAs IC Symposium, 1984.

[Mud91] T. N. Mudge and R. B. Brown, et. al, "The Design of a Micro-Supercomputer," *IEEE Computer Magazine*, pp. 57-64, 1991.

[Mun88] J. Munn, *GaAs Integrated Circuits*, 1st ed. New York: Macmillan Publishing Company, 1988.

[Nak85] N. Nakamura, "A 390ps 1000 gate array using GaAs super buffer FET logic," presented at ISSCC, 1985.

[Nak90] H. Nakano, M. Noda, M. Sakai, S. Matsue, and T. Oku, et. al, "A High-Speed GaAs 16Kb SRAM of 4.4ns/2W Using Triple-Level Metal Interconnection," presented at IEEE GaAs IC Symposium, 1990.

[Nak91] Y. Nakase, S. Kakutaro, K. Mashiko, and S. Kayano, "A 2-ns 16K Bipolar ECL RAM with Reduced Word-Line Voltage Swing," *IEEE Journal of Solid State Circuits*, vol. 26, pp. 518-524, 1991.

[Nam91] H. Nambu, et. al, "A 1.5ns, 64Kb ECL-CMOS SRAM," presented at 1991 Symposium on VLSI Circuits, 1991.

[Nam92] H. Nambu and K. Kazuo, et. al, "High-Speed Sensing Techniques for Ultra-high-Speed SRAM's," *IEEE Journal of Solid-State Circuits*, vol. 4, pp. 632-640, 1992.

[Neu91] P. G. Neudeck, et. al, "Significant Long-Term Reduction in n-Channel MESFET Subthreshold Leakage Using Ammonium-Sulfied Surface Treated Gates," *IEEE Electron Device Letters*, vol. 12, pp. 533-555, 1991.

[Num91] K. Nummila, et. al, "Short-Channel Effects in Sub-100nm GaAs MESFETs," *Electronic Letters*, vol. 27, pp. 1519-1521, 1991.

[OCo85] P. O'Conner, P. G. Flahive, and B. J. Roman, "A High Speed GaAs 1K Static Random Access Memory," *IEEE Journal of Solid State Circuits*, pp. 1080-1082, 1985.

[ONe93] P. O'Neil, B. Bernhardt, F. Nikpourian, C. Della, Y. Abad, G. Hansell, and G. Cooper, "GaAs Integrated Circuit Fabrication at Motorola," presented at IEEE GaAs IC Symposium, 1993.

[Ohb91] A. Ohba and S. Ohbayashi, et. al, "A 7-ns 1-Mb BiCMOS ECL SRAM with Shift Redundancy," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 507-512, 1991.

[Olu91] O. Olukotun, "Technology-Organization Tradeoffs in the Architecture of a High Performance Processor," : Michigan, 1991.

[Pas91] J. H. Pasternak and C. A. T. Salama, "GaAs MESFET Differential Pass-Transistor Logic," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1309-1316, 1991.

[Pu94] D. Pu and A. Gonzalez, "Testing of a GaAs MESFET Static RAM," University of Michigan, EECS 579 Report April 25 1994.

[Roc85] M. Rocchi, "Status of the surface and bulk effects limiting the performances of GaAs IC's," *Physica*, vol. 129 B, pp. 119-138, 1985.

[Roc86] C. e. a. Rocher, "Evaluation of the theoretical maximum fabrication of GaAs 1K bit SRAM's," presented at International Symposium on Gallium Arsenide and Rel. Compounds, 1986.

[Rya94] B. Ryan, "The New CPUs," in *Byte*, 1994, pp. 113-122.

[Sas92] K. Sasaki and K. Ishibashi, et. al, "A 7ns 140mW 1Mb SRAM with Current Sense Amplifier," presented at ISSCC, 1992.

[Sch91] N. Scheinberg and E. Chisholm, "A Capacitance Model for GaAs MESFET's," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1467-1470, 1991.

[Sch90]]D. Schmitt-Landsiedel, B. Hoppe, G. Nuendorf, M. Wurm, and J. Winnerl, "Pipeline Architecture for Fast CMOS Buffer RAM's," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 741-747, 1990.

[Sch92] S. Schuster, T. Chappell, B. Chappel, and R. Franch, "On-Chip Test Circuitry for a 2-ns Cycle, 512-kb CMOS ECL SRAM," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1073-1079, 1992.

[She86a] N. H. Sheng, H. T. Wang, C. P. Lee, G. J. Sullivan, and D. L. Miller, "A high speed 1 K-bit high electron mobility transistor static RAM," presented at Proceedings of GaAs IC symposium, 1986.

[She86b] N. H. Sheng, "A high speed 16 KBit HEMT SRAM," presented at GaAs IC Symposium, 1986.

[Shi90] S. Shimizu, Y. Kunio, and T. Terada, et. al, "An ECL-Compatible GaAs SCFL Design Methodology," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 539-545, 1990.

[Shin90] H. Shinohara, "A flexible mutli-port RAM compiler for datapath," presented at Proceedings of the 1990 Custom Integrated Circuits Conference, 1990.

[Shi91] H. Shinohara, N. Matsumoto, and K. Fujimori, et. al, "A Flexible Multiport RAM Compiler for Data Path," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 343-349, 1991.

[Sug86] T. Sugeta, T. Mizutani, M. Ino, and S. Horiguchi, "High speed technology comparison - GaAs vs Si," presented at GaAs IC Symposium, 1986.

[Suz91] M. Suzuki, et. al, "A 1.2-ns HEMT 64-kb SRAM," *IEEE Journal of Solid-State circuits*, pp. 1571-1576, 1991.

[Swa86] W. P. Swartz and e. al, "CMOS RAM, ROM and PLA Generators for ASIC Applications," presented at Proceedings of the 1986 Custom Integrated Circuits Conference, 1986.

[Sze81] S. M. Sze, *Physics of Semiconductor Devices*, 2nd ed. New York: Wiley, 1981.

[Tak90] M. Takada and N. Kazuyuki, et. al, "A 5-ns 1-Mb ECL BiCMOS SRAM," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1057-1062, 1990.

[Tak87] S. Takano, H. Makino, N. Tanino, M. Noda, K. Nishitani, and S. Kayano, "A GaAs 16Kbit static RAM," presented at ISSCC Technical Digest, 1987.

[Tak85] K. Takashi, T. Maeda, F. Katano, T. Furutsuka, and A. Higashisaka, "A CML GaAs 4 Kb SRAM," presented at ISSCC Digest of Technical Papers, 1985.

[Tan87] H. Tanaka, H. Yamasha, N. Masuda, N. Matsunaga, M. Miyazaki, H. Yanazawa, A. Masaki, and N. Hashimoto, "A 4K GaAs SRAM with 1ns access time," presented at ISSCC Digest of Technical Papers, 1987.

[Tan86] N. e. a. Tanino, "A 2.5ns 200mW GaAs 4Kb SRAM," presented at GaAs IC Symposium, 1986.

[Ter88] W. C. Terrell, C. L. Ho, and R. Hinds, "Direct Replacement of Silicon ECL and TTL SRAMs with High Performance GaAs Devices," presented at GaAs IC Symposium, 1988.

[Tou92] J. Tou, "A Submicrometer CMOS Embedded SRAM Compiler," *IEEE Journal of Solid State Circuits*, pp. 417-424, 1992.

[Toy86] N. Toyoda, K. Kanazawa, T. Terada, and M. Mochizuki, et. al, "A 256x4 Bit GaAs Static RAM," presented at GaAs IC Symposium, 1986.

[Tro83] G. L. Troeger and J. K. Nottoff, "A radiation-hard low-power GaAs static RAM using E-JFET DCFL," presented at GaAs IC Symposium, 1983.

[Tse87] C. T. Tsen, K. E. Kuwahara, and L. Salmon, et. al, "A Manufacturable Low-Power 16K-bit GaAs SRAM," presented at GaAs IC Symposium, 1987.

[Upt93] M. Upton and et. al, "A 160,000 transistor GaAs microprocessor," presented at 1993 IEEE International Solid-State Circuits Conference Digest of Technical Papers, 1993.

[Upt94] M. Upton, "A Latency Tolerant GaAs Microprocessor," PhD. Dissertation, University of Michigan, 1995.

[Vog88] C. Vogelsang, J. Castro, and J. Notthoff, et. al, "Complementary GaAs JFET 16K SRAM," presented at GaAs IC Symposium, 1988.

[Wad92] T. Wada, S. Rajan, and S. Przybylski, "An Analytical Access Time Model for On-Chip Cache Memories," *IEEE Journal of Solid-State Circuits*, vol. 8, pp. 1147-1156, 1992.

[Wil94] M. A. Wilson, Personal Communications, 1994.

[Yam92] K. Yamaguchi, H. Nambu, and Y. Kanetani, et. al, "A 1.5-ns Access Time, 78μm^2 Memory-Cell Size, 64-kb ECL-CMOS SRAM," *IEEE Journal of Solid State Circuits*, vol. 27, pp. 167-174, 1992.

[Yan85] L. Yang, A. T. Yen, and S. I. Long, "A simple method to improve the noise margin of III-V digital circuit coupling diode FET logic," *IEEE Electron device letters*, vol. EDL-7 (3), pp. 145-148, 1985.

[Yok84] N. Yokoyama, H. Onodera, T. Shinoki, H. Ohnishi, H. Nishi, and A. Shibatomi, "A 4Kx1b static RAM," presented at ISSCC Digest of Technical Papers, 1984.

[Yok85] N. e. a. Yokoyama, "A 3ns GaAs 4Kx1 bit static RAM," *IEEE Transactions on Electron Devices*, vol. 32, 1985.

[Yos84] K. e. a. Yoshihara, "Crosstalk predictions and reducing techniques for high speed GaAs digital IC's," presented at GaAs IC Symposium, 1984.

[Zde94] P. Zdebel, Personal Communications, 1994.